EXTENSIONS OF THE FORMLET MODEL OF PLANAR SHAPE

ALEXANDER YAKUBOVICH

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ARTS

GRADUATE PROGRAM IN MATHEMATICS AND STATISTICS
YORK UNIVERSITY
TORONTO, ONTARIO

OCTOBER 2014

Abstract

This thesis addresses the problem of shape representation using the GRID/formlet theory, a system based on localized diffeomorphisms. While this framework enjoys many desirable properties, it suffers from several limitations: it converges slowly for shapes with elongated parts, and it can be sensitive to parameterization as well as grossly ill-conditioned. Several innovations are proposed to address these problems: 1) The formlet basis is generalized to include oriented deformation, improving convergence for elongated parts. 2) A recent contour remapping method is applied in order to eliminate problems due to drift of the model parameterization during matching pursuit. 3) A regularizing term is introduced in order to limit redundancy in formlet parameters and improve the model's identifiability. Finally, an algorithm is proposed to hierarchically cluster formlets, and is shown to induce a partial ordering on the representation.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

Shape representation is an important part of many computer vision problems, including shape segmentation, recognition, and tracking [26]. While there is a rich literature on shape representation from the points of view of biological vision, wavelet analysis, and statistics, few methods respect the topology of shape. A notable exception is the GRID/formlet model, which explains variation in shapes by means of localized diffeomorphisms of the plane [16, 26]. While the aim of GRID is to model an observed biological growth process, formlets model an observed shape through a latent growth process that begins with some embryonic shape (e.g. an ellipse).

This thesis aims to make the formlet representation a more efficient code for natural shapes. First, oriented deformations are introduced into the formlet basis, improving the speed at which shapes can be reconstructed. Second, a novel parameterization method based on optimal contour mappings [23] is implemented, further improving convergence. Third, by defining a regularizing term in the data space, the energy of deformations used to represent a given shape is reduced. Finally, a graphical model is formulated in order to better capture the hierarchical nature of animal shapes and to induce a partial ordering on the formlet code.

I begin by motivating the problem of shape representation and reviewing prior work.

1.1 What is shape?

Three-dimensional objects project to two-dimensional silhouettes. If the objects are simply-connected and do not self-occlude, the boundaries of their silhouettes will be simple (nonintersecting), closed curves.

This thesis will only consider planar shapes that are smooth, simple, closed curves; $\Gamma : [0, 1] \rightarrow \mathbb{R}^2, \quad \Gamma(0) = \Gamma(1)$. Let Ω_Γ be the space of all shapes.

Two very simple representations of shape are the binary image of the region $S = interior(\Gamma)$, or a vector of points sampled along Γ. These representations are limited because they fail to satisfy a minimal set of properties listed in section 1.2. This thesis will consider the problem of building a shape representation satisfying these essential properties. Throughout this work animal shapes will be used as a representative case.

1.2 Essential properties of a shape model

Elder et al. identify a minimal set of properties a shape model must satisfy [12]:

1. *Completeness* - Any shape $\Gamma \in \Omega_\Gamma$ can be represented by the model.

2. *Closure* - Ω_Γ is closed under the generative model; that is, sampling from the model only yields shapes in Ω_Γ.

3. *Composition* - A complex shape is represented as a composition of simpler components.

4. *Sparsity* - Accurate approximations of shape can be generated with relatively few components.

5. *Progression* - Incorporating more components improves reconstruction accuracy.

6. *Locality* - Components are localized in space.

7. *Scaling* - Components have different scales and are self-similar over scale.

8. *Region & Contour* - Components capture both region and contour properties.

I claim that two more properties must be added before the representation can be successfully incorporated in a vision system:

1. *Identifiability* - A statistical model is *identifiable* if changing the value of a model parameter always generates a different probability distribution [8]. This condition must be satisfied for learning to be tractable, that is, for the estimated parameters to converge to their true values. This property also encompasses the notion of stability, that is, that small perturbations in the input data result in small changes in the representation. Humans are able to recognize objects even when subjected to different sources of variations (noise in the imaging system, change in perspective, movement, variation within a class) [24]. For a vision system to be robust to these variations, the representation it uses must be stable.

2. *Part-structure* - There exists a natural correspondence between components of the model and what people identify as parts of a shape. This allows for a more structured, hierarchical representation that can be used for object recognition and other tasks.

Lastly, the shape model should be generative, as described in the following section.

1.3 Generative models

A common problem in statistics and machine learning is inference about some hidden world state W given observed data D. Modelling approaches generally fall into one of two categories [28]:

1. *Generative models*, which specify the joint distribution $P(W, D)$. With generative models it is possible to sample any variable in the model.

2. *Discriminative models*, which only specify the conditional distribution $P(W|D)$, without describing the distribution of the observed variables.

In the context of shape modelling, W is a set of latent variables encoding the observed shape D. Discriminative models are sufficient for a number of prediction tasks. However, many applications in computer vision require a more flexible generative model which assigns probabilities to different shapes. These include:

1. Segmentation, recognition, and tracking in cluttered scenes. The task at hand is not only to discriminate shapes from one another, but also from 'phantom' shapes formed by features from multiple objects and the background.

2. Modelling of shape articulation, growth and deformation.

3. Modelling shape similarity

2 Literature review

Shape models can be categorized as either: 1) contour-based models, which are only sensitive to $\Gamma = \partial S$, and 2) symmetry-based models, which model the interior of S through its symmetry axes.

2.1 Contour-based models

Contour-based models typically define a family of basis functions that are used to describe shape. A given curve is represented as a linear superposition of basis functions. There are different choices of basis functions, including the spline bases, Fourier descriptors, and Principal Component Analysis (PCA), representations amongst others.

2.1.1 Principal Component Analysis (PCA) bases

PCA bases describe the statistics of landmarks within an object class [9]. Landmarks are features on an object (e.g. points on the shape) that correspond across multiple instances of the object. These can be domain-specific (e.g. eyes in a face) or domain-independent (e.g. curvature extrema). A third type of landmark can be formed by interpolating the points between two other landmarks.

To compare the statistics of landmarks from different shapes, the shapes must be in the same coordinates system. This can be achieved by Procrustes transformation, defined as the scaling, rotation, and translation that minimizes the sum of squared differences across corresponding points on two shapes. Cootes et al. use a generalized form of the Procrustes transformation where the residual error is weighted by the stability of the landmarks, defined as the inverse of their total variance across exemplars (the idea being that landmarks that are more stable across different shapes are more important) [9].

Following the notation of [9], let $\mathbf{x}_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1} \dots x_{in-1}, y_{in-1})^T \in \mathbb{R}^{2n}$ be a vector representing the i^{th} shape using n points in the plane. A set of N observed shapes can be thought of as a point cloud in $\mathbb{R}^{2n}$ which can be described by its first two moments. The centroid of the cloud is given by the mean across all N shapes: $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$. The covariance matrix is a $2n \times 2n$ matrix given by:

$$S = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x_i} - \overline{\mathbf{x_i}})^T(\mathbf{x_i} - \overline{\mathbf{x_i}}) \tag{1}$$

Principal components are defined as the unit eigenvectors of this matrix, that is, the vectors $\mathbf{p}_1 \dots \mathbf{p}_{2n}$ such that, for $i = 1 \dots 2n$,

$$S\mathbf{p}_i = \lambda_i \mathbf{p}_i \tag{2}$$

$$\mathbf{p}_i^T \mathbf{p}_i = 1 \tag{3}$$

The eigenvectors define the major axes of the point cloud, assuming it is ellipsoidal. They can be interpreted as the "modes of variation" in the shape, that is, directions along which the projection of the data has the greatest variance. Often much of the variance in the data can be explained using a few eigenvectors, say $K < 2n$ eigenvectors, leading to a sparse representation. Let $\mathbf{b}_k$ be the k^{th} eigenvector represented in complex coordinates. A given shape $\Gamma(t)$ can be written as:

$$\Gamma(t) = \Gamma_0(t) + \sum_{k=0}^{K} a_k \mathbf{b}_k(t) + \mathbf{e}(t) \tag{4}$$

where Γ_0 is the centroid of the shape, $\{a_k\}_{k=1}^K$ are the weights, and $\mathbf{e}$ is the vector of residual error between the observed shape and its projection onto the PCA basis. The PCA representation has proven useful in several areas of application, most notably in medical image analysis [10]. It provides a rich statistical model of variation within an object class that is robust to linear transformations of the data, as well as noise and occlusion. However, it suffers from two main limitations. First, the basis functions are global, making the representation sensitive to occlusion. If a small part of an observed shape is corrupted, the representation will be changed drastically. Second, it requires ground truth data on landmarks, which is expensive and hard to establish for some object categories.

2.1.2 Fourier descriptors

Fourier representations use the family of trigonometric basis functions [30]:

$$\left\{ \frac{1}{2\pi}, \frac{\cos t}{\pi}, \frac{\sin t}{\pi}, \frac{\cos 2t}{\pi}, \frac{\sin 2t}{\pi} \dots \right\} \tag{5}$$

Letting $t = \left\{ \frac{2\pi}{n} \right\}, n = 0, 1 \dots N$, a sampled shape Γ can be expressed as:

$$\Gamma(t) = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=0}^{K} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos \pi k t \\ \sin \pi k t \end{bmatrix} + \mathbf{e}(t) \tag{6}$$

where $\mathbf{e}(t)$ is the residual error. The Fourier basis enjoys the property of orthonormality, meaning

$$\begin{cases} \langle \phi_i, \phi_j \rangle = 0 & \forall i, j = 1 \dots 2K+1 \\ \|\phi_i\|_2 = \sqrt{\langle \phi_i, \phi_i \rangle} = 1 & \forall i = 1 \dots 2K+1 \end{cases} \tag{7}$$

where the ϕ_i are the basis functions $\phi_k = (\cos \pi kt, \sin \pi kt)$ and the inner product of ϕ_i and ϕ_j is defined as:

$$\langle \phi_i, \phi_j \rangle := \int_0^1 \phi_i(t)\phi_j(t)dt \tag{8}$$

Orthonormality minimizes redundancy and makes learning the representation more tractable. While Fourier representations have found considerable success in the area of medical image analysis (e.g. [30]) , they fall short of providing a complete generative model of shape for two reasons. First, as in the PCA basis, the basis functions are global - each describes variation across the entire shape (albeit at different frequencies). Second, while Fourier representations provide an explicit parametric model, the relationship between model parameters and shape features is not very clear.

2.1.3 Basis-splines

The main limitation of Fourier and PCA bases is that they are global. One family of basis functions that overcomes this limitation is the basis-splines (B-splines). Splines are piecewise polynomials that are constrained to be smooth where the the pieces meet. B-splines are splines of minimal degree, meaning that if the domain $[0, 1]$ is partitioned into K intervals,

$$\pi = (t_0, t_1 \dots t_K), \quad t_0 < t_1 < \dots < t_K \text{ where } t_0 = 0, t_K = 1 \tag{9}$$

a B-spline of degree d will span exactly d-intervals. Letting $B_k^d(t; \pi)$ be a B-spline of degree d, (x_k, y_k) be a control point, and $\mathbf{e}$ be a vector of residual error, the representation can be written as:

$$\Gamma(t) = \sum_{k=1}^{K} \begin{bmatrix} x_k \\ y_k \end{bmatrix} B_k^d(t; \pi) + \mathbf{e} \tag{10}$$

A big limitation of the spline model is that it is highly sensitive to the partition π. While each basis function is localized, adding or removing a spline changes the partition and in doing so, affects adjacent bases. Also, using a finer partition decreases the support of existing bases, so the model is confined to describing shape at a single scale.

For these reasons, most modern approaches to shape representation capture information at multiple scales.

2.1.4 Curvature-based models

The curvature function $\kappa(s)$ is a complete description of a shape $\Gamma(s)$. Any planar curve can be reconstructed from the curvature function, up to rotation and translation, though this method is highly unstable [24].

Attneave [2] proposed a shape representation based on curvature extrema, where information is concentrated. Curvature can also be used to define parts of a shape - Hoffman and Richards [17] segment a contour into parts by finding curvature minima.

2.1.5 Shapelets

Dubinsky and Zhu move a step closer towards a complete generative model in their shapelet representation [11]. Similar to other contour-based methods, a shape is represented by a linear combination of basis functions. However, their approach is unique in that the basis functions are highly localized and capture information at different scales. In addition, their model is truly linearly additive, in the sense that existing bases are not perturbed when a new term is added or removed (c.f. splines). The shapelet basis consists of Gabor-like planar curves $\psi : [0, 1] \to \mathbb{C}$ parameterized by location μ along arc length and scale σ:

$$\psi(t; \mu, \sigma) \propto e^{-\frac{(t-\mu)^2}{2\sigma^2}} \left(\cos\left(\frac{2\pi}{\sigma}(t-\mu)\right) + i \sin\left(\frac{2\pi}{\sigma}(t-\mu)\right)\right) \tag{11}$$

The basis functions are subject to an affine transformation $z \mapsto Az + z_0$, where $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is the matrix of basis coefficients. The affine transformation allows anisotropic scalings, rotations and shears.

In the signal processing literature, a *dictionary* is an exhaustive set of elementary signals called atoms. When there are more atoms than possible signals (meaning that there is more than one way to represent a given signal), the dictionary is said to be *over-complete*. Discretizing the parameter space (A_k, μ_k, σ_k) yields an overcomplete dictionary of shapelets.

Shapelets can be composed to represent arbitrarily complex shapes:

$$\Gamma^K(t) \approx \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \sum_{k=1}^{K} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \psi(t; \sigma_k, \mu_k) \tag{12}$$

where $(x_0, y_0)^T$ is the centroid of the given contour.

Learning the representation is harder than for the case of Fourier or PCA bases, where the projection of the signal onto the basis is unique. The authors build on the the matching pursuit algorithm that is widely used for wavelet coding [22]. Matching pursuit is a greedy algorithm, meaning that it iteratively finds

locally optimal solutions. At each iteration, one finds the projection onto the dictionary which minimizes reconstruction error, measured by the $\mathcal{L}^2$ norm of the residual. Since the real and imaginary components are orthogonal with respect to the inner product $\langle f, g \rangle = \int_0^1 f(x)g(x)dx$, the optimal linear transformation can be found analytically if the scale σ and the position μ are known. This means that the dictionary only needs to be discretized along two dimensions, not six, making the optimization problem very tractable.

The shapelet model has many strengths. It is a multi-scale, sparse representation with localized components. However, it suffers from two limitations: 1) it does not capture regional properties of shape, and 2) the topology of object boundaries is not captured in the model (i.e. random samples of the model may have self-intersecting boundaries). Therefore, the model fails to satisfy the crucial closure property outlined in Section 1.2.

2.2 Symmetry-based models

Blum et al. introduced the symmetry axis model of shape, in which planar shapes are represented in terms of two 1-dimensional functions: a skeleton function and a radius function [4] [5]. This approach exploits the symmetries present in natural shapes. It led to the smoothed local symmetries (SLS) model and related representations which found applications in medical imaging and other domains [6].

Subsequent work introduced the notions of scale and time into the symmetry axis representation. Leyton introduced the idea that every shape can be modelled by a latent growth process, with symmetry axes defining the directions for growth [19], [20]. He defined a set of rules, called the Process Grammar, to qualitatively infer the deformation history for a given shape. A key result in this framework is the symmetry-curvature duality theorem, which establishes a one-to-one correspondence between curvature extrema and symmetry axes.

Building on these ideas, Feldman and Singh introduced a Bayesian probabilistic model for computing the skeleton in which shapes are 'grown' from their skeletons through a stochastic generative process [13]. By introducing a prior over the curvature and number of branches in the skeleton, they overcome the problem of spurious branches which plagued earlier deterministic algorithms, leading to a more stable representation. While Bayesian estimation improves the compactness of the skeleton, the representation is still not sparse, and may require more storage than the original contour.

2.3 Hybrid models

In [35], Zhu proposes a shape representation that combines both region and contour based properties in a natural way. The model is a Markov Random Field whose neighbourhood structure corresponds to both

contour-based and region-based Gestalt laws (collinearity, co-circularity and proximity, parallelism and symmetry respectively). One promising aspect of this approach is that it provides a complete probabilistic model, which allows one to sample the learned probability distribution $P(\Omega)$ over the space of natural shapes. However, there are two main limitations to this theory: (I) It is not sparse, and is in fact higher-dimensional than the given contour, (II) the topological constraint is not embedded in the model in a natural way - a 'firewall' is employed during Markov Chain Monte Carlo sampling to reject shapes that self intersect. This is inefficient, and creates a disconnect between the sampling distribution and the generative variables used in the model.

2.4 Coordinate transformations

Another class of model, which can also be thought of as region-based, uses transformations of the image space to understand the contour. D'Arcy Thompson pioneered this idea in his seminal work *On Growth and Form*, in which he used coordinate transformations to model morphological changes of animal species over time [31].

Extending these ideas to the field of computer vision, Jain et al. were amongst the first to use a Fourier deformation basis to match observed shapes to prototypes [18]. The Fourier basis, however, is not local, and fails to satisfy the closure property.

More recently, Sharon and Mumford explored conformal mappings as global coordinate transformations between shapes [29]. The Reimann mapping theorem guarantees that the unit circle can be conformally mapped to any simple closed curve. Such mappings do not generally preserve the topology of embedded contours, and therefore the set of valid shapes is again not closed under these transformations.

In considering prior models of shape, the most difficult goal to attain seems to be that of closure: ensuring the models only generate valid shapes. While the coordinate transformations considered so far do not preserve the topology of embedded contours, it is possible to define a family of transformations that will. Constraining the transformations to be diffeomorphisms, that is – smooth, invertible mappings – will yield a generative model that satisfies the closure property.

2.4.1 Growth by random iterated diffeomorphisms (GRID)

Grenander et al. propose using localized coordinate transformations to model growth of anatomical parts [15], [16]. The transformations are constrained to be diffeomorphisms, defined as follows:

Definition 1. *A* **diffeomorphism** f *is a smooth, one-to-one (injective) and onto (surjective) mapping* $f : \mathbb{R}^n \to \mathbb{R}^n$.

Intuitively, if the image plane is thought of as a rubber sheet, a diffeomorphism is a transformation consisting of bending or twisting of the sheet without tearing it or folding it onto itself. Grenander et al. view biological growth as a composition of elementary deformations of the image plane, accumulated over time. The deformations are parametric radial deformation fields centred at randomly sampled seeds. Parameter estimation is done in two stages: (1) the cumulative deformation map is estimated, and (2) the individual GRID variables are estimated from the deformation map. In more recent work, the GRID variables are estimated directly from the image data. This is done using an unconstrained optimal control problem expressed in a contour-based coordinate system [27]. The original formulation assumed the deformations have no spatial interactions, yielding an additive model. This assumption is relaxed in later work where a gradient-based energy minimization is used to iteratively estimate the GRID variables [33].

Earlier approaches fall short of providing a complete generative model of shape because they fail to satisfy the closure property - arbitrary combinations of components do not always result in valid shapes. By using diffeomorphisms as basis functions, the GRID model overcomes this limitation. Since the topology of a shape can only change if it is subject to a non-injective or discontinuous transformation, it follows that a diffeomorphism always preserves the topology of the embedded shape. Furthermore, diffeomorphisms form a group under composition, meaning that an arbitrary composition of diffeomorphisms is also a diffeomorphism. It follows that the topology of an object is invariant under an arbitrary sequence of diffeomorphisms.

3 Formlets

While the GRID model is promising in many respects, it does not actually address the problem of shape representation, but rather, the related problem of modelling of differential growth (i.e. decomposing an observed evolution of a shape into localized deformations). Building on this work, the formlet theory uses localized diffeomorphisms to represent shape with a generative model [26], [12]. While much of the underlying mathematics is similar, the fundamental difference is that the growth process is observed in the GRID framework but is latent with Formlets.

Starting with an embryonic shape Γ^0, diffeomorphisms can be composed to represent an arbitrarily complex shape:

$$\Gamma^K(t) = (f_K \circ f_{K-1} \circ \ldots \circ f_1)(\Gamma^0(t))$$

The family of diffeomorphisms consists of Gabor-like deformations localized in scale and space. They define a *formlet* as the isotropic, radial (angle-preserving) mapping $f : \mathbb{R}^2 \to \mathbb{R}^2$,

$$f(\mathbf{x}; \mathbf{x_0}, \sigma, \alpha) = \mathbf{x_0} + \frac{\mathbf{x} - \mathbf{x_0}}{\|\mathbf{x} - \mathbf{x_0}\|}\rho(\|\mathbf{x} - \mathbf{x_0}\|, \theta), \quad \text{where} \tag{13}$$

$$\rho(r; \alpha, \sigma) = r + \alpha \sin\left(\frac{2\pi r}{\sigma}\right) \exp\left(\frac{-r^2}{\sigma^2}\right) \tag{14}$$

This transformation can also be expressed as a perturbation added to the identity:

$$f(\mathbf{x}; \mathbf{x_0}, \sigma, \alpha) = \mathbf{x} + \alpha g(\mathbf{x} - \mathbf{x_0}; \sigma), \quad \text{where} \tag{15}$$

$$g(\mathbf{x} - \mathbf{x_0}; \sigma) = \frac{\mathbf{x} - \mathbf{x_0}}{r} \sin\left(\frac{2\pi r}{\sigma}\right) \exp\left(\frac{-r^2}{\sigma^2}\right) \tag{16}$$

Each formlet can be thought of as a packet of three parameters: I) location $\mathbf{x_0} \in \mathbb{R}^2$, II) scale $\sigma \in \mathbb{R}^+$, and III) gain $\alpha \in \mathbb{R}$. The scale controls the size of the region affected by a formlet, that is, the degree of localization. The gain controls the magnitude and sign of the perturbation. Figure 1 shows the composition of four formlets chosen at random.

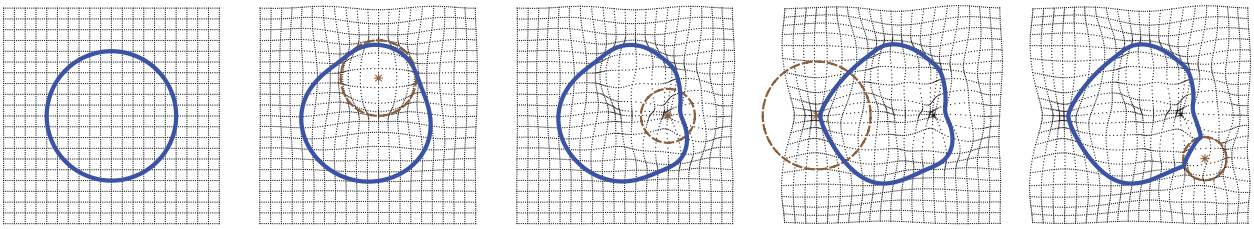
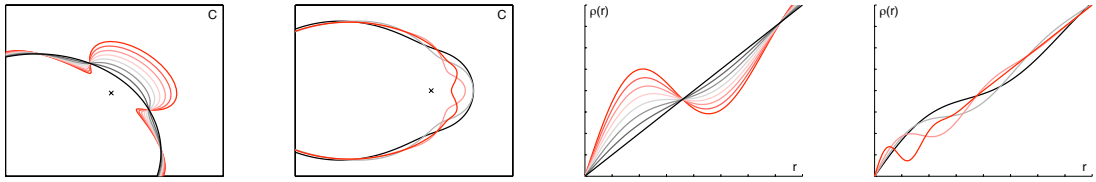


Figure 1: Sequence of formlets applied to an embedded circle. The asterisk indicates the location parameter $\mathbf{x_0}$, and the dotted circle indicates the scale σ (for all points $\mathbf{x}$ on the dotted circle, $\|\mathbf{x} - \mathbf{x_0}\| = \sigma$). The formlet parameters are sampled from a uniform distribution. Taken from [26].

Figure 2 shows the effect of each parameter, both in terms of the function $\rho(r)$ and the deformation applied to a contour.



(a) ρ with gain variation (b) ρ with scale variation (c) f with gain variation (d) f with scale variation

Figure 2: Figures (a) and (b) show several formlets applied to an ellipse as the gain (α) and scale (σ) parameters are varied, and Figures (c) and (d) show their associated deformation functions $\rho(r)$. The formlets shown in red and orange do not satisfy the diffeomorphism constraints (that is, $\rho(r)$ is decreasing for some r). Taken from [26].

A careful examination of Equation 13 will reveal that the transformation is in fact not always injective. However, a simple linear constraint on the gain parameter, namely, $-\frac{\sigma}{2\pi} \leq \alpha \leq .1956\sigma$, guarantees that it is.

The constraint is derived in [26] by noting that that since the angle of a point relative to the seed $\mathbf{x_0}$ is fixed by the mapping, it suffices to show that the radial deformation function is monotonically increasing .

3.1 Computing the representation

There are two parts to learning the formlet representation $(f_K \circ f_{K-1} \ldots \circ f_1)(\Gamma_0(t))$ of a given shape: defining the embryonic shape Γ_0, and inferring the formlet parameters $\{f_1 \ldots f_K\}$.

3.1.1 Initialization

To initialize the model, Oleskiw et al. first normalize the target shape, represented as a $n = 128$ point polygon, to have zero mean and unit variance in the horizontal and vertical directions:

$$\int_0^1 \Gamma_x(t)dt = \int_0^1 \Gamma_y(t)dt = 0 \tag{17}$$

$$\int_0^1 \Gamma_x(t)^2dt = \int_0^1 \Gamma_y(t)^2dt = 1 \tag{18}$$

This normalization guarantees the scales of the shapes will be on the same order of magnitude. It is also claimed in [26] that the normalization is required for the ellipse initialization to be computed analytically, but in section 6.5 it will be shown that this is not the case.

They then sample the unit circle at $n = 128$ evenly spaced points, and form a 1:1 correspondence between the model and target points that remains fixed throughout the pursuit [26]. The unit circle is then translated by the centroid of the target contour, and subjected to a linear transformation yielding an ellipse minimizing the $\mathcal{L}^2$ error between the two curves.

3.1.2 Formlet pursuit

Oleskiw et al. use a greedy algorithm to find the formlet sequence $\{f_1 \dots f_K\}$ representing an observed shape. In each iteration k,the formlet f_k maximally reducing reconstruction error is selected [12], [26]:

$$f_k = \arg\min_f E(\Gamma^{obs}, f(\Gamma^k - 1)) \tag{19}$$

Error is measured by the $\mathcal{L}^2$ norm of the residual:

$$E(\Gamma^{obs}, \Gamma^K) = \|\Gamma^{obs} - \Gamma^K\|_2 \tag{20}$$

$$= \int (\Gamma_x^{k-1}(t)) - \Gamma_x^{obs}(t))^2 + (f(\Gamma_y^{k-1}(t)) - \Gamma_y^{obs}(t))^2 dt \tag{21}$$

where Γ^{obs} is the observed shape vector, Γ^k is its representation at the kth iteration, and Γ_x, Γ_y specify the x and y coordinates of a shape Γ. Experimentation with gradient descent revealed that the search space in this non-convex optimization problem is riddled with local minima [26]. One approach to solving the problem [26] is to adopt the matching pursuit strategy of using a dictionary of candidate deformations [22], replacing linear combination of basis elements with non-commutative composition. Discretizing the parameter space $(x_0, y_0, \sigma, \alpha) \in \mathbb{R}^2 \times \mathbb{R}^+ \times \mathbb{R}$ yields an over-complete dictionary $\mathcal{D}$ which can be searched exhaustively.

Fortunately, the dimensionality of the search space can be reduced by noting that formlets are linear in

the gain parameter α. Hence, the reconstruction error is a quadratic function of the gain and so there exists a closed form solution for the optimal gain (given the remaining parameters). To derive this solution, first note that the residual error at each iteration k can be written as:

$$\|\Gamma^{obs} - f(\Gamma^{k-1}; \alpha, \sigma, \mathbf{x_0})\|_2^2 = \|\Gamma^{obs} - \Gamma^{k-1} - \alpha g(\Gamma^{k-1} - \mathbf{x_0})\|_2^2 \tag{22}$$

$$= \|\Gamma^{res} - \alpha g(\Gamma^{k-1} - \mathbf{x_0})\|_2^2 \tag{23}$$

$$= \|\Gamma^{res}\|_2^2 + \alpha^2 \|g\|_2^2 - 2\alpha \langle \Gamma^{res}, g \rangle \tag{24}$$

where $\Gamma^{res} := \Gamma^{obs} - \Gamma^{k-1}$, $g := g(\Gamma^{k-1} - \mathbf{x_0}; \sigma)$ and $\langle \Gamma^a, \Gamma^b \rangle := \int_0^1 \Gamma_x^a(t)\Gamma_x^b(t) + \Gamma_y^a(t)\Gamma_y^b(t)dt \quad \forall \Gamma^a, \Gamma^b : [0,1] \to \mathbb{R}^2$. Differentiating with respect to α and setting to zero:

$$0 = \frac{\partial}{\partial \alpha} \left(\|\Gamma^{res}\|_2^2 + \alpha^2 \|g\|_2^2 - 2\alpha \langle \Gamma^{res}, g \rangle \right) \tag{25}$$

$$= 2\alpha \|g\|_2^2 - 2 \langle \Gamma^{res}, g \rangle \tag{26}$$

This yields the following expression for the optimal gain:

$$\alpha^* = \frac{\langle \Gamma^{res}, g \rangle}{\|g\|_2^2} \tag{27}$$

The dictionary search method can be further refined by using a sparser dictionary but deploying gradient descent from the most promising formlets. This so-called dictionary-descent method allows for faster and more accurate shape reconstruction [12].

This thesis will present four extensions of the formlet theory of planar shape:

1. Generalizing the formlet basis to include oriented deformations which are well suited to describing elongated parts of natural shapes.

2. Optimally adapting the parameterization of the shape representation, allowing for faster shape reconstruction.

3. Regularizing the inverse problem of shape reconstruction, leading to to a more identifiable and stable model.

4. Hierarchically clustering formlets, leading to a model that captures some of the part structure of natural shapes. This structured representation may serve as the foundation for a probabilistic graphical model of a shape.

The extended formlet model will be compared against the original formlet model as as well the shapelet model on two tasks: convergence rate for shape reconstruction, and contour completion.

4 Evaluation

4.1 Dataset

Throughout this work, a dataset of 252 animal images selected from the Hemera Photo-Object database will be used [26]. The images are blue-screened: the alpha channel of each image specifies the object boundary. Figure 3 shows a sample of six images from the dataset.

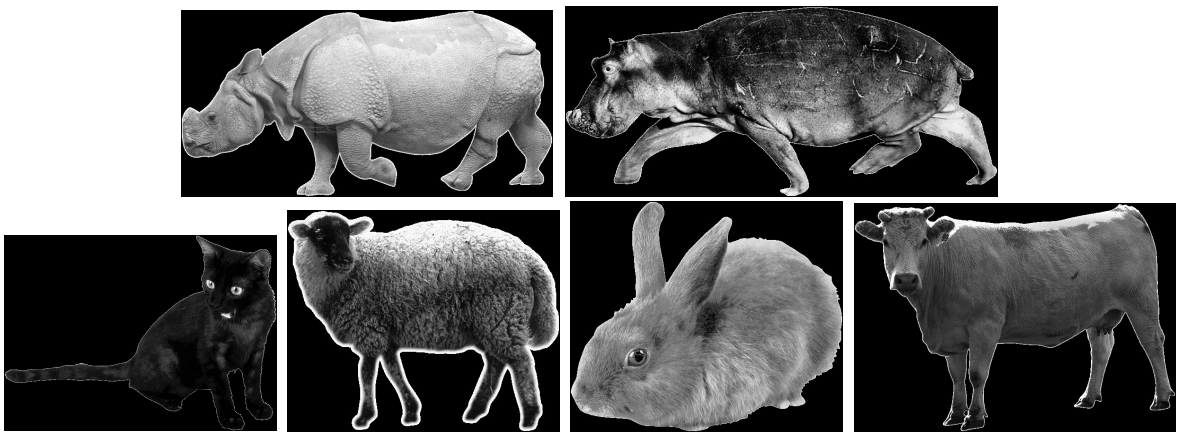


Figure 3: Sample of images from the Hemera object database.

The boundaries are represented as vectors of $n = 128$ points sampled at equal arc length intervals. Increasing the resolution improves the approximation but also increases the storage and runtime for the reconstruction algorithms.

4.2 Shape Reconstruction

First, the models will be evaluated on the task of shape reconstruction. Since the models are progressive, with enough basis elements they can encode a given contour to an arbitrary precision. The rates of convergence, however, will vary depending on how well the models capture features of natural shapes. The error between the model and target contours will be measured by the $\mathcal{L}^2$ Hausdorff error, which is chosen because it is independent of the parameterizations of the two curves. This metric defines the error between the target and model shapes as the symmetrical average minimum distance of a point on one shape to the other:

$$\xi^H(\Gamma^{obs}, \Gamma^k) = \frac{1}{2}\left(\overline{\min_{t' \in [0,1)} d\left(\Gamma^{obs}(t), \Gamma^k(t')\right)} + \overline{\min_{t' \in [0,1)} d\left(\Gamma^{obs}(t'), \Gamma^k(t)\right)}\right) \tag{28}$$

where $d(\mathbf{x}, \mathbf{y})$ denotes the Euclidean distance between vectors $\mathbf{x}$ and $\mathbf{y}$ and $\overline{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$ denotes the mean over all points on the shape.

4.3 Contour completion

The second problem is contour completion, which arises naturally when object boundaries in an image are fragmented. As shown in Figure 4, this can happen because the objects are occluded (amodal completion) or because there is not enough contrast between figure and ground (modal completion).

Figure 4: Object boundaries project to the image as fragmented contours, due to occlusions (cyan) and low figure/ground contrast (red). This gives rise to the problems of amodal and modal completion, respectively. Taken from [32].

Contour completion is the process of filling in the missing parts. It is an important part of a perceptual organization framework, where edges need to be grouped into closed contours. Completions can be used to to assign probabilities to different groupings, and guide the search for additional evidence.

5 Generalizing the formlet basis

5.1 Motivation

The family of deformations presented in [12], [26] was restricted to isotropic mappings. In such mappings, the amount by which a point is displaced depends only on its distance from the centre of the formlet, not the angle. Consider a set of points S_c that are equidistant from the formlet centre $\mathbf{x_0} : S_c = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x} - \mathbf{x_0}\|_2 = c\}$ for some $c > 0$. Every point in S_c will be displaced by exactly the same amount.

While isotropic deformations are good for representing shapes with roughly circular features, the reconstruction algorithm converges slowly for shapes with elongated parts, such as the limbs of the giraffe shown in Figure 5.

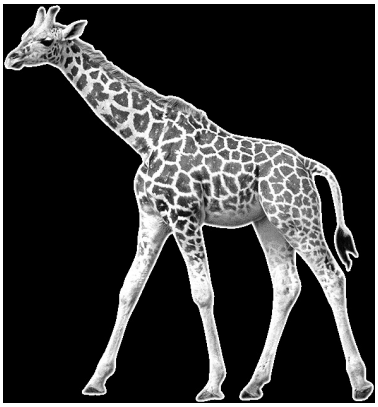


Figure 5: Giraffe image from the Hemera object dataset.

By generalizing the basis to include oriented deformations, it may be possible to improve the convergence rate and attain more localized deformations that have clear association with parts of a shape.

5.2 Oriented formlets

The method I will propose is inspired by the approach used by Grenander et al. in their GRID model [15], [16]. They defined a rich family of deformations using radial and angular deformation functions. They used an exponential form for the radial deformation function, $R(r) = re^{-\left(\frac{r}{c}\right)^2}$, and a nonparametric form for the angular deformation function (ADF). This allowed the number of maxima and minima of the ADF to vary, which potentially results in multimodal deformations (e.g. a unipolar deformation with one axis along which the deformation acts, and similarly, bipolar and multipolar deformations).

Instead, I consider a lower dimensional model based upon the von Mises distribution [14]. The von Mises

distribution $A(\theta; \theta_0, \kappa)$ is a circular function on $\theta \in \mathbb{S}$. It is defined as follows:

$$A(\theta; \theta_0, \kappa) = \frac{\exp(k \cos(\theta - \theta_0)}{2\pi I_0(\kappa)} \tag{29}$$

Note that $A(\theta) = A(\theta + 2\pi) \quad \forall \theta \in \mathbb{S}$. θ_0 is a location parameter and κ is a scaling parameter controlling the spread about the mode (for $\kappa = 0$, it is uniformly constant). Figure 6 shows the von Mises function for different values of the concentration parameter κ.

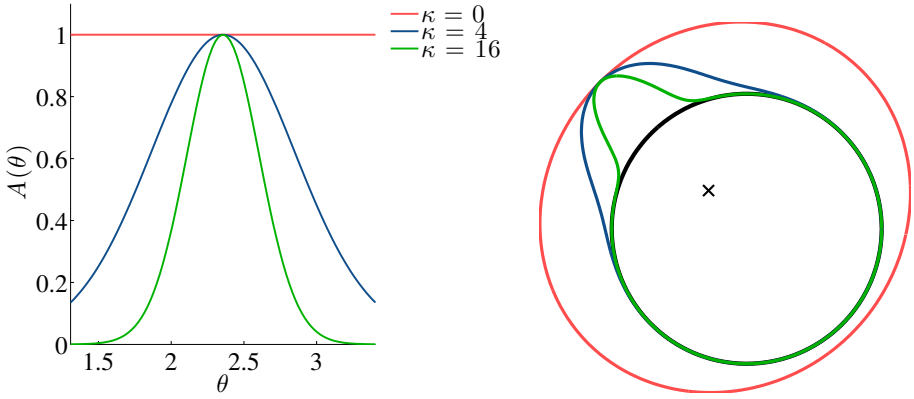


Figure 6: Effect of the concentration parameter κ on the von Mises angular deformation function (left) and on a formlet applied to the unit circle (right). The x indicates the location parameter $\mathbf{x_0}$.

5.3 Oriented formlet parameterization

A point $\mathbf{x} \in \mathbb{R}^2$ can be expressed in a polar coordinate system centred at $\mathbf{x_0}$ (i.e. $(x, y) \leftrightarrow (r, \theta)$, where $r = \|\mathbf{x} - \mathbf{x_0}\|$ and $\theta = \arg(\mathbf{x} - \mathbf{x_0})$). Since formlets are radial (angle-preserving) transformations, it follows that:

$$\begin{aligned} \mathbf{x} \mapsto f(\mathbf{x}) \iff \\ (r, \theta) \mapsto (\rho(r, \theta), \theta) \end{aligned} \tag{30}$$

where $\rho : \mathbb{R}^+ \times \mathbb{S} \mapsto \mathbb{R}^+$. The function ρ can be factored into radial and angular deformation functions. As in [26], [12], the radial deformation function $R(r)$ is a Gabor, that is, a sinusoid attenuated by an exponential function. The angular deformation function $A(\theta)$ is a von Mises function. Thus, an oriented formlet can be written as:

$$f(\mathbf{x}; \mathbf{x_0}, \sigma, \alpha, \kappa, \theta_0) = \mathbf{x_0} + \frac{\mathbf{x} - \mathbf{x_0}}{\|\mathbf{x} - \mathbf{x_0}\|} \rho(\|\mathbf{x} - \mathbf{x_0}\|, \theta), \quad \text{where} \tag{31}$$

$$\rho(r,\theta) = r + \alpha \cdot R(r;\sigma) \cdot A(\theta;\theta_0,\kappa) \tag{32}$$

$$R(r;\sigma) = \sin\left(\frac{2\pi r}{\sigma}\right)\exp\left(\frac{-r^2}{\sigma^2}\right) \tag{33}$$

$$A(\theta;\theta_0,k) \propto \exp(\kappa\cos(\theta-\theta_0)) \tag{34}$$

The transformation in Equation (31) can be expressed as as a perturbation added to the identity, where the gain parameter α controls the magnitude of the perturbation:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{x_0} + \frac{\mathbf{x}-\mathbf{x_0}}{r}\left[r + \alpha A(\theta)R(r)\right] \\ &= \mathbf{x} + \frac{\mathbf{x}-\mathbf{x_0}}{r}\alpha A(\theta)R(r) \\ &:= \mathbf{x} + \alpha g(\mathbf{x}-\mathbf{x_0}), \quad \text{where } g(\mathbf{x}-\mathbf{x_0}) = \frac{\mathbf{x}-\mathbf{x_0}}{r}A(\theta)R(r) \end{aligned} \tag{35}$$

Figure 6 shows oriented formlets with different concentration parameters applied to a circle.

Next, I consider the problem of constraining oriented formlets to be diffeomorphic.

Proposition 1. *A formlet f is guaranteed to be a diffeomorphism if $\frac{-\sigma}{2\pi} < \alpha < .1956\sigma$ and $0 \le A(\theta) \le 1 \quad \forall\theta \in [0, 2\pi]$*

Proof. Since f is radial, it will be injective if and only if $\rho(r,\theta)$ is monotonically increasing or decreasing. ρ cannot be everywhere decreasing, since $\rho(0,\theta) = 0$ and ρ is everywhere nonnegative. Therefore, it remains to find the conditions under which $\rho(r,\theta)$ is monotonically increasing:

$$\frac{\partial}{\partial r}\rho(r,\theta) > 0 \quad \forall\theta \in [-\pi,\pi], r \in [0,\infty) \tag{36}$$

From Equation 32 it follows that:

$$\frac{\partial}{\partial r}\rho(r;\theta) = 1 + \alpha A(\theta)R'(r;\theta) \tag{37}$$

$$= 1 + \alpha A e^{-r^2/\sigma^2}\left[\frac{2\pi}{\sigma}\cos\left(\frac{2\pi r}{\sigma}\right) - \frac{2r}{\sigma^2}\sin\left(\frac{2\pi r}{\sigma}\right)\right] \tag{38}$$

$$\equiv 1 + \alpha A\psi(r) \tag{39}$$

where $\psi(r) := e^{-r^2/\sigma^2}\left[\frac{2\pi}{\sigma}\cos\left(\frac{2\pi r}{\sigma}\right) - \frac{2r}{\sigma^2}\sin\left(\frac{2\pi r}{\sigma}\right)\right]$. Assume that $0 \le A(\theta) \le 1$. If $\alpha < 0$:

$$0 < 1 + \alpha A\psi(r) \tag{40}$$

$$< 1 + \alpha\psi(0), \quad \text{since } \arg\max_{r>0}\psi(r) = 0 \tag{41}$$

$$= 1 + \alpha A \frac{2\pi}{\sigma} \tag{42}$$

$$\iff -\frac{\sigma}{2\pi} < \alpha A, \quad \text{for } 0 \leq A(\theta) \leq 1 \tag{43}$$

$$\iff -\frac{\sigma}{2\pi} < \alpha \tag{44}$$

If $\alpha > 0$:

$$0 < 1 + \alpha A \psi(r) \tag{45}$$

$$0 < 1 + \alpha A \psi_{min}, \quad \text{for } 0 \leq A(\theta) \leq 1 \tag{46}$$

$$0 < 1 + \alpha \psi_{min} \tag{47}$$

$$\alpha < \frac{-1}{\psi_{min}} \approx \frac{-1}{-5.112/\sigma} = 0.1956\sigma \tag{48}$$

where it can be shown numerically that $\psi_{min} := \min_r \psi(r) = -5.112/\sigma$. □

Since $A(\theta)$ is everywhere nonnegative, to guarantee that $|A(\theta)| \leq 1$ everywhere it suffices to normalize by the mode, $A(\theta_0) = \exp(\kappa \cos(0)) = \exp(\kappa)$. Therefore,

$$A(\theta; \theta_0, k) = \exp(-\kappa) \exp(\kappa \cos(\theta - \theta_0)) \tag{49}$$

$$= \exp(\kappa \cos(\theta - \theta_0) - \kappa) \tag{50}$$

5.4 Results

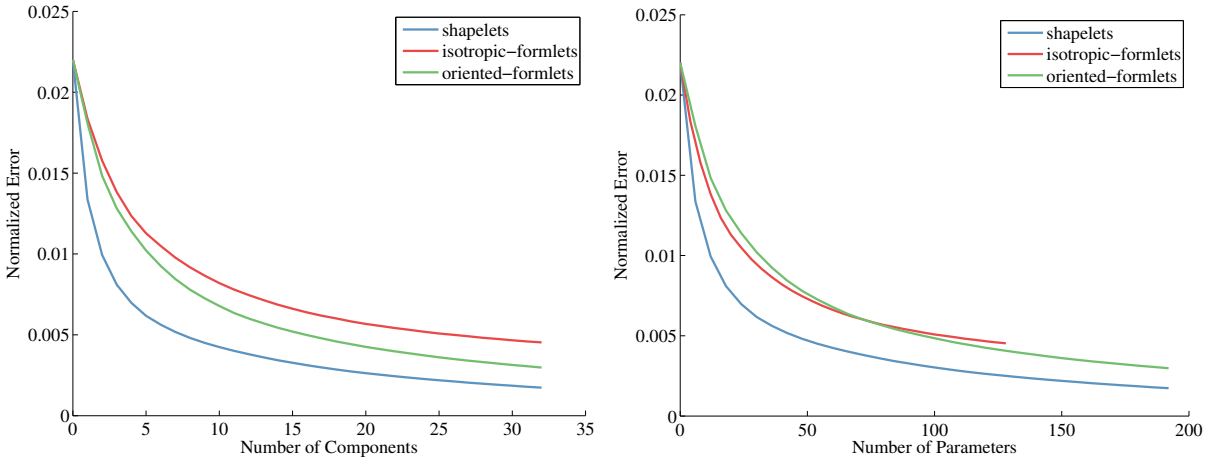
The three generative models of shoe presented so far – shapelets, isotropic formlets, and oriented formlets – can now be compared. Figure 8 shows the pursuit of an example giraffe, and Figure 7 shows quantitative results on the shape reconstruction task. The models have different dimensionality - shapelets have six parameters per basis element $\{\mu, \sigma, a, b, c, d\}$, isotropic formlets have four $\{x_0, y_0, \sigma, \alpha\}$, and oriented formlets have six $\{x_0, y_0, \sigma, \alpha, \theta_0, \kappa\}$. Therefore, to evaluate the sparsity of the representations, both the number of basis elements and the total number of parameters need to be considered (Figure 7a and Figure 7b respectively) . Several observations can be made:

1. Shapelets converge faster than formlets. This may be because they do not respect topological constraints, or in the case of comparing to istotropic formlets, because they have more parameters.

2. Convergence with respect to the number of basis elements is faster for oriented formlets than for isotropic formlets. Interestingly, considering convergence with respect to the total number of parameters, isotropic formlets outperform oriented formlets until the two models have 72 parameters (iteration

18 for isotropic formlets, iteration 12 for oriented formlets), at which point oriented formlets start to do better.

3. The shapelet approximation has several topological errors, that is, self-intersections. However, these are corrected as the algorithm converges.



(a) Convergence – number of basis elements.

(b) Convergence – total number of parameters.

Figure 7: Convergence with shapelets, isotropic and oriented formlets. Note that shapelets and oriented formlets have six parameters per basis elements, while isotropic formlets have four.

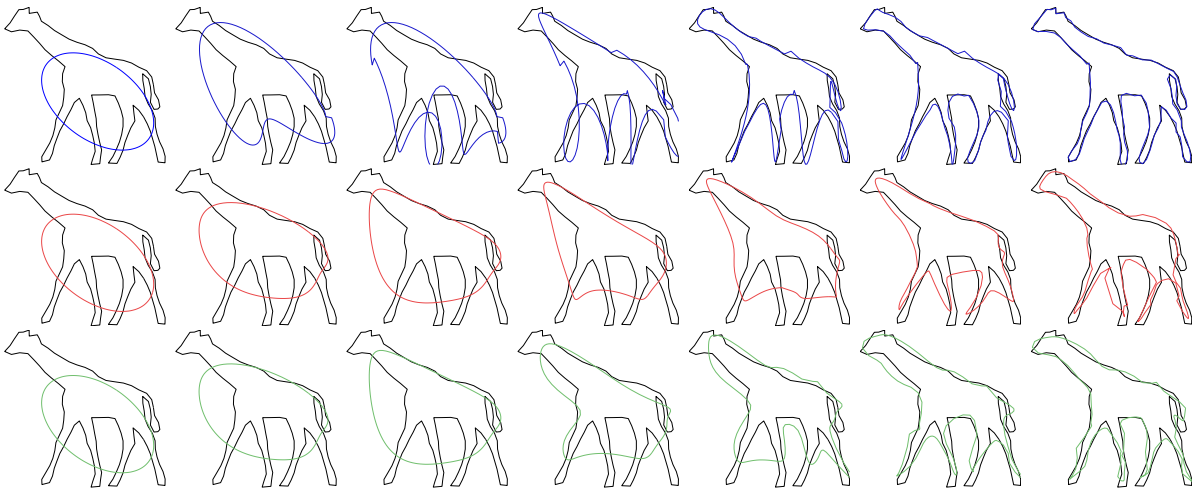


Figure 8: Pursuit of giraffe shape with shapelets (top row), isotropic formlets (middle row) and oriented formlets (bottom row). Iterations 0,1,2,4,16,32 are shown from left to right.

6 Adapting the parameterization of the shape representation

6.1 Motivation

Fitting a model to a given shape requires a measure of similarity between the target and representation. In the formlet system, similarity between shapes Γ^A and Γ^B is measured by the $\mathcal{L}^2$ norm of the residual:

$$E(\Gamma^A, \Gamma^B) = \|\Gamma^A - \Gamma^B\|_2^2 \tag{51}$$

$$= \sum_{i=1}^{n} (\Gamma_x^A(t_i) - \Gamma_x^B(t_i))^2 + (\Gamma_y^A(t_i) - \Gamma_y^B(t_i))^2 \tag{52}$$

This objective function assumes that the correspondence between points on the two contours has been established. There are many possible mappings between the curves, and the correspondence will affect the representation that is learned. This chapter considers the problem of adapting the parameterization of the model curve to optimize correspondence between the two curves, and the related problem of finding the best ellipse initialization.

As discussed in Section 3.1.1, Oleskiw et al. [26] initialized the pursuit with a unit circle centred at the centroid of the target shape, and subject to a linear transformation minimizing squared error between the two curves. The residual error is based on a correspondence between the two contours in which the first point on the circle (that is, $\cos(0), \sin(0)) = (1, 0)$) is mapped to the first point on the target shape. The contours have the same number of points and are both oriented counter-clockwise, so matching the first points uniquely determines the correspondence.

One problem with the correspondence is that it does not evolve together with the model contour and thus will generally no longer be optimal as the model contour is deformed (i.e it will produce a much larger residual than a Hausdorff distance, and could lead to slower convergence and the selection of unnatural formlets). In addition to being based on a poor correspondence, the initialization is suboptimal because it fixes the ellipse centre instead of searching for the best value along with the linear transformation parameters.

To solve these problems, I propose a method for reparameterizing the model shape to establish an accurate 1:1 mapping with the target shape. This method is then used to compute an optimal ellipse initialization, making sure to search over all the parameters of the affine transformation.

6.2 The contour mapping measure

Movahedi et al. considered the problem of measuring the correspondence between two contours in the context of evaluating segmentation algorithms [23]. They considered a number of methods for measuring the distance

between two simple, closed curves, and demonstrated that the methods used extensively in the literature exhibit serious failure modes. Based upon this analysis, they proposed a novel *contour mapping measure* for measuring the distance between two curves, and showed that it is more consistent with human judgement of shape similarity than competing methods.

Their results are summarized below, adapting the notation to be consistent with that used throughout this thesis. Let $\Gamma^A = a_1 a_2 \dots a_m$, $\Gamma^B = b_1 b_2 \dots b_n$ and let $s : a \leftrightarrow b$ define a mapping between points a and b on the two contours. The mapping sequence $S = s_1 \dots s_k$ is defined as a mapping between Γ^A and Γ^B in which every point on each curve is mapped to at least one point on the other. The cost of the mapping is defined as:

$$\gamma(S) = \sum_{i=1}^{k} \gamma(s_i) \tag{53}$$

where $\gamma(s_i)$ is the Euclidean distance between the matched points. Movahedi et al. [23] argue that the mapping should respect the arc length ordering of the two contours, and therefore constrain the mapping to be monotonic:

$$\text{If } a_i \leftrightarrow b_m, a_j \leftrightarrow b_n \text{ then } i < j \Rightarrow m \leq n \text{ and } m < n \Rightarrow i \leq j \tag{54}$$

The mapping distance between the two curves is defined as:

$$\delta(\Gamma^A, \Gamma^B) := \min_S \gamma(S) \tag{55}$$

and the optimal mapping between them is:

$$S^*(\Gamma^A, \Gamma^B) := \arg\min_S \gamma(S) \tag{56}$$

Mappings can be visualized as paths in a directed graph, as shown in Figure 9. The mapping is a path from $(1,1)$ to (m,n), where at each vertex (i,j), there are three ways to grow the path: match a_i to b_{j+1} (move right), match a_{i+1} to b_j (move down), or match a_{i+1} to b_{j+1} (move diagonally). The problem of computing the optimal path decomposes into overlapping subproblems as follows. Let $\Gamma_i^A = a_1 \dots a_i$ and $\Gamma_j^B = b_1 \dots b_j$. For $i = j = 1$, we have that $\delta(\Gamma_1^A, \Gamma_1^B) = d(a_1, b_1)$. For $i, j > 1$, the cost of mapping Γ_i^A to

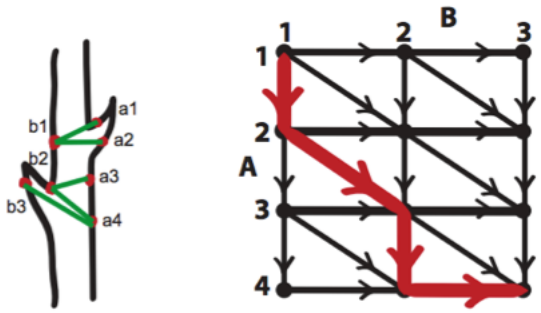


Figure 9: Example of a mapping between contours Γ^A and Γ^B. Taken from [23].

Γ^B_j can be written recursively as:

$$\delta(\Gamma^A_i, \Gamma^B_j) = d(a_i, b_j) + \min \begin{cases} \delta(\Gamma^A_{i-1}, \Gamma^B_{j-1}) \\ \delta(\Gamma^A_{i-1}, \Gamma^B_j) \\ \delta(\Gamma^A_i, \Gamma^B_{j-1}) \end{cases} \tag{57}$$

In other words, the cost of the path from $(1,1)$ to (i,j) is the cost of matching $a_i \leftrightarrow b_j$ plus the lowest cost of the path to one of its predecessors. Given this decomposition, the optimization problem can be solved efficiently using dynamic programming (the dynamic programming table can be built with complexity $O(n)$).

Throughout this derivation, it has been assumed that $a_1 \leftrightarrow b_1$. Since the curve are closed, all possible cyclic permutations must be considered (e.g. $\sigma^k(a_1 \dots a_n) = a_{k+1} \dots a_n a_1 \dots a_k$). The best permutation is given by:

$$k^* := \arg\min_k \delta(\sigma^k(\Gamma^A), \Gamma^B) \tag{58}$$

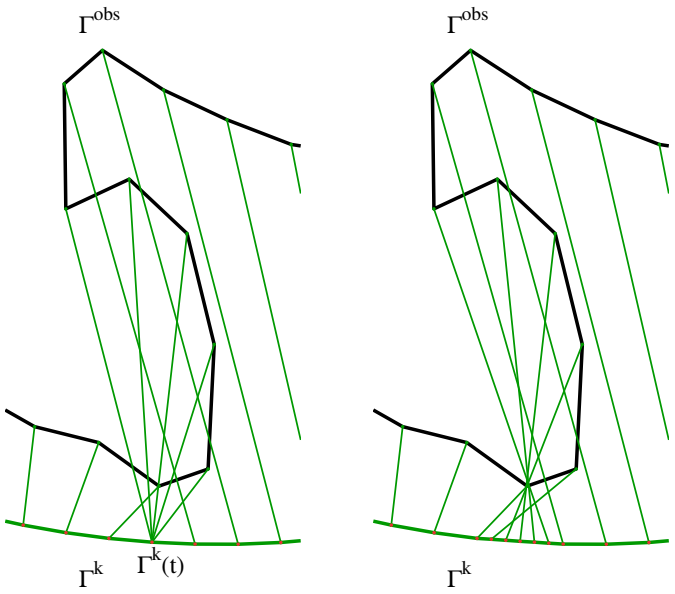
Finally, the optimal mapping between Γ^A and Γ^B can be expressed as:

$$S^*(\sigma^{k^*}(\Gamma^A), \Gamma^B) \tag{59}$$

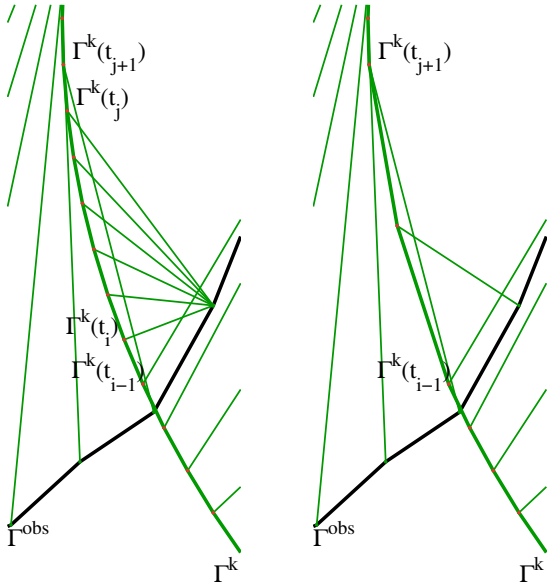
Constructing the dynamic programming table for all m cyclic shifts has complexity $O(m^2 n)$, but it can be improved to $O(nm \log m)$ by exploiting the cyclic structure of the problem [21].

6.3 Re-parameterization

The Contour Mapping Measure allows for one-to-many and many-to-one correspondences between the two contours. Figure 10 shows an example of each case. These matchings occur when there there is a different



(a) A one-to-many matching (left), corrected by introducing points as needed (right).



(b) A many-to-one matching (left), corrected by removing points as needed (right).

Figure 10: Re-parameterizing the model contour to guarantee a one-to-one correspondence. Γ^{obs}, shown in black, is the observed contour and Γ^k, shown in green, is the model.

degree of detail in the parts being matched (e.g. a circular arc on the model shape is matched to an elongated limb on the target shape).

It is critical that the measure have this flexibility, otherwise the correspondence between the two shapes will become misaligned. However, a 1:1 correspondence is also crucial for formlet pursuit, otherwise convergence may stall. For example, consider a a case where a point on the pursuing contour is matched to two

points that are equidistant but in opposite directions. Suppose that the residual on the rest of the contour is zero. Then there is no formlet that can reduce the residual, since any step that decreases the error on one point will necessarily increase the error on the other. Thus, the algorithm will be stuck in a fixed point.

To address this problem, after optimizing the model parameterization with the contour mapping measure, a 1:1 mapping is restored with a series of split and merge operations on the model points. There are two steps:

1. *(One-to-many)* If a point $\Gamma^k(t)$ on the model curve is matched to n points on the target curve with $n > 1$, introduce n points on the model by linearly interpolating between the neighbours of $\Gamma^k(t)$.

2. *(Many-to-one)* If n points $\Gamma^{k-1}(t_i) \ldots \Gamma^{k-1}(t_j)$ on the model curve are matched to a single point on the target curve, replace the n points by the midpoint of the two points flanking the n points, $\Gamma^k(t_{i-1}), \Gamma^k(t_{j+1})$.

6.4 Evaluation

The adaptive reparameterization method is evaluated with respect to the rate of convergence on the database of animal shapes, as in Section 5. Figure 11 shows that optimizing the correspondence with the CM measure seems to improve convergence substantially.

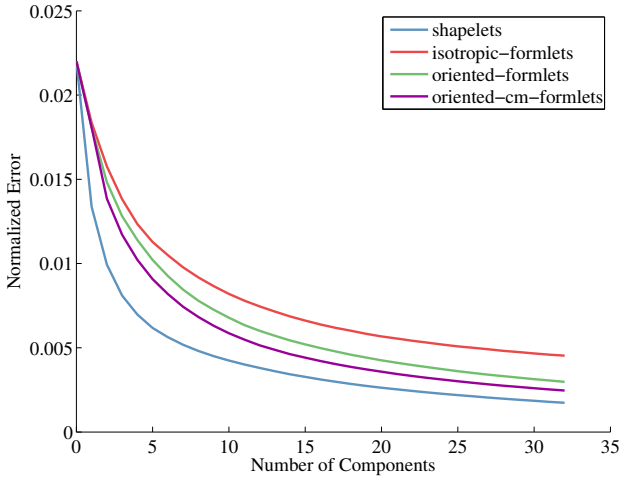


Figure 11: Comparing mean convergence rates for shapelets, isotropic formlets, and oriented formlets (with and without reparameterization).

6.5 Ellipse Initialization

Having defined a method for parameterizing the model contour to have a natural correspondence with the target, I now address the problem of optimizing the ellipse used to initialize the pursuit.

To find the best-fitting ellipse, the Principal Components of the given shape (viewed as a point cloud in $\mathbb{R}^2$) are computed. I find the ellipse whose principal axes are aligned with the eigenvectors of the covariance matrix of the shape and whose lengths are set to the square roots of the corresponding eigenvalues (in a similar vein to the PCA bases described in section 2.1.1).

The correspondence is then refined using the Contour Mapping and re-parameterization methods described in sections 6.2 and 6.3. Having computed an accurate correspondence, the two shapes are registered using an affine transformation. This involves finding the affine transformation that, when applied to the model contour, minimizes the $\mathcal{L}^2$ residual error with respect to the learned correspondence.

It is straightforward to derive an analytical solution to this least squares problem. The squared $\mathcal{L}^2$-residual error between the two contours is given by:

$$E(A, b; \Gamma^{target}, \Gamma_0) = \|A\Gamma_0 + b - \Gamma^{target}\|_2^2 \tag{60}$$

$$= \sum_{i=1}^{n} \|A\mathbf{x}_i + b - \mathbf{x}_{target}\|^2 \tag{61}$$

$$= \sum_{i=1}^{n} (ax_i + by_i + b_x - x_i^{obs})^2 + (cx_i + dy_i + b_y - y_i^{obs})^2 \tag{62}$$

Since the error is a quadratic function of the affine parameters, the problem reduces to setting the gradient to zero and finding the roots:

$$0 = \frac{\partial E}{\partial a} = \sum_{i=1}^{n} 2(ax_i + by_i + b_x - x_i^{obs})x_i \tag{63}$$

$$= a\sum_i x_i^2 + b\sum_i x_i y_i + b_x \sum_i x_i - \sum_i x_i x_i^{obs} \tag{64}$$

$$0 = \frac{\partial E}{\partial b} = \sum_{i} 2(ax_i + by_i + b_x - x_i^{obs})y_i \tag{65}$$

$$= a\sum_i x_i y_i + b\sum_i y_i^2 + b_x \sum_i y_i - \sum_i y_i x_i^{obs} \tag{66}$$

$$0 = \frac{\partial E}{\partial c} = \sum_{i}^{n} 2(cx_i + dy_i + b_y - y_i^{obs})x_i \tag{67}$$

$$= c\sum_i x_i^2 + d\sum_i x_i y_i + b_y \sum_i x_i - \sum_i x_i y_i^{obs} \tag{68}$$

$$0 = \frac{\partial E}{\partial d} = \sum_i^n 2(cx_i + dy_i + b_y - y_i^{obs})y_i \tag{69}$$

$$= c\sum_i x_i y_i + d\sum_i y_i^2 + b_y \sum_i y_i - \sum_i y_i y_i^{obs} \tag{70}$$

$$0 = \frac{\partial E}{\partial b_x} = \sum_i^n 2(ax_i + by_i + b_x - x_i^{obs}) \tag{71}$$

$$= a\sum_i x_i + b\sum_i y_i + nb_x - \sum_i x_i^{obs} \tag{72}$$

$$0 = \frac{\partial E}{\partial b_y} = \sum_i^n 2(cx_i + dy_i + b_y - y_i^{obs}) \tag{73}$$

$$= c\sum_i x_i + d\sum_i y_i + nb_y - \sum_i y_i^{obs} \tag{74}$$

This is a system of six linear equations which can be expressed in matrix form as:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & 0 & 0 & \sum x_i & 0 \\ \sum x_i y_i & \sum y_i^2 & 0 & 0 & \sum y_i & 0 \\ 0 & 0 & \sum x_i^2 & \sum x_i y_i & 0 & \sum x_i \\ 0 & 0 & \sum x_i y_i & \sum y_i^2 & 0 & \sum y_i \\ \sum x_i & \sum y_i & 0 & 0 & n & 0 \\ 0 & 0 & \sum x_i & \sum y_i & 0 & n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ b_x \\ b_y \end{bmatrix} = \begin{bmatrix} \sum x_i x_i^{obs} \\ \sum y_i x_i^{obs} \\ \sum x_i y_i^{obs} \\ \sum y_i y_i^{obs} \\ \sum x_i^{obs} \\ \sum y_i^{obs} \end{bmatrix} \tag{75}$$

The vector of optimal affine parameters is the solution to this linear system of equations. After the affine transformation is applied to the model, the correspondence determined with CM is generally no longer optimal. Therefore, the steps of parameterizing the ellipse and updating the ellipse parameters with an affine transformation are iterated. Algorithm 1 summarizes the steps described thus far.

Algorithm 1: Fitting an ellipse to an observed shape

Initialization: Fit PCA ellipse Γ_0

for $j = 1, \ldots, J$ **do**

UPDATE CORRESPONDENCE: CM and re-parameterization

UPDATE ELLIPSE PARAMETERS: Apply an affine transformation minimizing $\mathcal{L}^2$ error:

$$\Gamma^j = A\Gamma^{j-1} + b$$

where $\{A, b\} = \arg\min_{\{A,b\}} \|A\Gamma^{j-1} + b - \Gamma^{obs}\|$

Note that a justification for normalizing the shapes was so that the ellipses could be fitted analytically, and it has been shown that this is not necessary. Normalization also ensures that the shapes will have similar scales, however, it distorts the aspect ratio of the shapes, and measurements on the distorted shapes will not generally carry over to the original shapes (i.e. the residual error for a model will not be the same in the two coordinate systems). To avoid these complications, original image coordinates will be used from now on.

7 Regularizing the inverse problem

7.1 Motivation

The formlet system models shape through a sequence of deformations of the space it is embedded in. The objective function used to select the deformations, however, is only sensitive to properties of the bounding contour.

This is problematic because two very different formlets may have the same effect on the contour. Consider the pursuit of the cat shape shown in Figure 12. It appears that a formlet with a scale on the order of 10^6 pixels has a similar effect on the embryonic ellipse as a formlet with a scale on the order of 10^2 pixels. This compromises the stability and the identifiability of the model, and makes it more difficult to interpret its parameters.

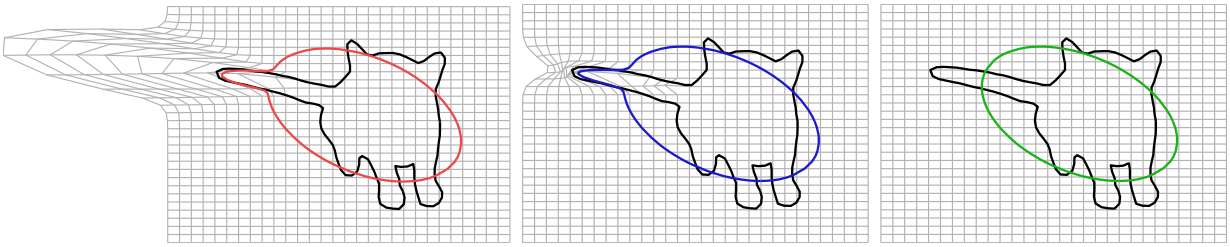


Figure 12: Two formlets applied to the ellipse initialization (shown in green). The scales of the formlets shown in red and blue are 3.1×10^6 pixels and 730 pixels respectively. The image is 1465×682 pixels.

7.2 Regularized objective function

The problem can be regularized by adding a penalty to the objective function. Given a formlet with parameters $\mathbf{p} = (x_0, y_0, \sigma, \alpha, \kappa, \theta_0)$, the objective function E can be written as a weighted sum of a fidelity term F and a penalty term G:

$$E(\Gamma^{obs}, f(\Gamma^{k-1}; \mathbf{p})) = F(\Gamma^{obs}, f(\Gamma^{k-1}; \mathbf{p})) + \lambda G(\mathbf{p}) \tag{76}$$

The selection of a formlet becomes a compromise between the improvement in reconstruction error and the strength of the penalty. Regularization is equivalent to introducing a nonuniform prior distribution over the formlet parameters (for example, a prior belief that large-scale formlets deforming space outside of the image are unlikely).

One way to define the regularizing term would be to add a norm of the vector of formlet parameters $\|\mathbf{p}\|$ to the energy function, as with the ridge and LASSO penalties in the context of regression analysis [3]. A

possible problem with this formulation is that the penalties on the formlet parameters are independent, which may not be reasonable. It is also unclear how to weight the penalties for each of the different parameters.

Instead of working the parameter space, the penalty may be defined directly in the data space. The penalty G is defined as the energy of the deformation field, which is obtained by integrating the magnitude of the deformation over the plane:

$$G(\mathbf{p}) = \int_{\mathbb{R}^2} \|f(\mathbf{x};\mathbf{p}) - \mathbf{x}\|_2^2 \quad d\mathbf{x} \tag{77}$$

To evaluate this integral, first recall the parameterization of an oriented formlet:

$$f(\mathbf{x}) = \mathbf{x_0} + \frac{\mathbf{x}-\mathbf{x_0}}{r}\left[r + \alpha A(\theta)R(r)\right], \quad r = \|\mathbf{x}-\mathbf{x_0}\|, \quad \theta = \arg(\mathbf{x}-\mathbf{x_0}) \tag{78}$$

$$= \mathbf{x} + \frac{\mathbf{x}-\mathbf{x_0}}{r}\alpha A(\theta)R(r), \quad \text{where} \tag{79}$$

$$R(r;\sigma) = \sin\left(\frac{2\pi r}{\sigma}\right)\exp\left(\frac{-r^2}{\sigma^2}\right) \qquad \text{(radial deformation function)} \tag{80}$$

$$A(\theta;\theta_0,\kappa) = \exp(\kappa\cos(\theta-\theta_0)-\kappa) \qquad \text{(angular deformation function)} \tag{81}$$

The squared displacement of a point $\mathbf{x} \in \mathbb{R}^2$ is given by:

$$\|f(\mathbf{x}) - \mathbf{x}\|_2^2 = \|\mathbf{x} + \frac{\mathbf{x}-\mathbf{x_0}}{r}\alpha A(\theta)R(r) - \mathbf{x}\|_2^2 \tag{82}$$

$$= (\alpha A(\theta)R(r))^2 \cdot \|\frac{\mathbf{x}-\mathbf{x_0}}{r}\|_2^2 \tag{83}$$

$$= (\alpha A(\theta)R(r))^2 \tag{84}$$

To find the energy of a formlet, integrate the displacement over the image plane:

$$G = \int_{r=0}^{\infty}\int_{-\pi}^{\pi} (\alpha A(\theta)R(r))^2 r d\theta dr \tag{85}$$

$$= \alpha^2 \int_{-\pi}^{\pi} A^2(\theta)d\theta \int_0^{\infty} R^2(r)rdr \tag{86}$$

The integral over θ can be solved analytically:

$$\int_{\theta=0}^{2\pi} A^2(\theta)d\theta = \int_{-\pi}^{\pi} (\exp(\kappa\cos\theta-\kappa))^2 d\theta \tag{87}$$

$$= e^{-2\kappa}\int_{-\pi}^{\pi}\exp(2\kappa\cos\theta)d\theta \tag{88}$$

$$= 2\pi I_0(2\kappa)e^{-2\kappa}\int_{-\pi}^{\pi}\frac{1}{2\pi I_0(2\kappa)}\exp(2\kappa\cos\theta)d\theta \tag{89}$$

$$= 2\pi I_0(2\kappa)e^{-2\kappa} \tag{90}$$

Note that Equation 89 assumes the following normalization identity for the von Mises function [14]:

$$\int_{\theta=-\pi}^{\pi} \frac{1}{2\pi I_0(\kappa)} \exp(\kappa \cos\theta) = 1 \quad \forall \kappa > 0 \tag{91}$$

Note that $I_\alpha(\kappa)$ is the modified Bessel function:

$$I_\alpha(\kappa) = \sum_{m=0}^{\infty} \frac{1}{m! \cdot \Gamma(m+\alpha+1)} \left(\frac{\kappa}{2}\right)^{2m+\alpha} \tag{92}$$

$$\text{Thus,} \quad I_0(\kappa) = \sum_{m=0}^{\infty} \frac{1}{m! \cdot \Gamma(m+1)} \left(\frac{\kappa}{2}\right)^{2m} \tag{93}$$

$$= \sum_{m=0}^{\infty} \left(\frac{1}{m!}\left(\frac{\kappa}{2}\right)^m\right)^2 \tag{94}$$

where $\Gamma(m+1) = m!$. Here Γ is the gamma function, an extension of the factorial function to real and complex numbers.

To find the integral over r, substitute: $u = \dfrac{r}{\sigma} \Rightarrow r = u\sigma \Rightarrow dr = \sigma du$:

$$\int_{r=0}^{\infty} |R(r)|^2 r dr = \int_{r=0}^{\infty} \left|\exp\left(\frac{-r^2}{\sigma^2}\right) \sin\left(\frac{2\pi r}{\sigma}\right)\right|^2 r dr \tag{95}$$

$$= \int_{u=0}^{\infty} \left|\exp(-u^2)\sin(2\pi u)\right|^2 (u\sigma)\sigma du \tag{96}$$

$$= \sigma^2 \int_{0}^{\infty} \left|\exp(-u^2)\sin(2\pi u)\right|^2 u du \tag{97}$$

$$\approx 0.1284\sigma^2 \tag{98}$$

The last step was done numerically using an adaptive Simpson quadrature method. Substituting these expressions back into Equation 86 yields the following expression for Equation 77:

$$G = \alpha^2 \int_{-\pi}^{\pi} A^2(\theta) d\theta \int_{0}^{\infty} R^2(r) r dr \tag{99}$$

$$\approx \alpha^2 \cdot 2\pi I_0(2\kappa)e^{-2\kappa} \cdot 0.1284\sigma^2 \tag{100}$$

$$= C\alpha^2\sigma^2 I_0(2\kappa)e^{-2\kappa}, \quad \text{where } C \approx 0.2568\pi \tag{101}$$

Note that this penalty is very different from a simple norm of the formlet parameters. First, the location and orientation parameters have been integrated out, which makes sense since they do not affect the energy. Next, the effect of the penalty on each parameter depends on the value of the others, and the effect is

highly nonlinear. For example, increasing scale by $\Delta\sigma$ increases the penalty G by $C\alpha^2(\Delta\sigma)^2 I_0(2\kappa)e^{-2\kappa}$. The penalty is quadratic in the scale and gain, and negative exponential in κ (the exponential term $e^{-2\kappa}$ dominates the Bessel function term $I_0(2\kappa)$ for large κ). Thus, formlets with large scales and gains but low concentrations are most heavily penalized, which is desirable since they are very global and therefore more sensitive to occlusions.

Note that the fidelity term is a line integral over the shape whereas the penalty term is an area integral over the plane. To express the two terms in the same units, the fidelity term is multiplied by a constant L approximating the area of the residual: $R = (S^{obs} \cup S^k) - (S^{obs} \cap S^k)$, where $S^{obs} = \text{interior}(\Gamma^{obs})$ and $S^k = \text{interior}(\Gamma^k)$:

$$F = L\sum_{i=1}^{n}(f(\Gamma_x^{k-1}(t_i)) - \Gamma_x^{obs}(t_i))^2 + (f(\Gamma_y^{k-1}(t_i)) - \Gamma_y^{obs}(t_i))^2, \quad L = l_1 l_2 \tag{102}$$

where l_1 is the average length of line segments on the observed and model contours and l_2 is the average residual from the previous iteration:

$$l_1 = \frac{1}{2}\left[\frac{1}{n}\sum_{i=1}^{n}\|\Gamma^{k-1}(t_i) - \Gamma^{k-1}(t_{i-1}))\|_2 + \frac{1}{n}\sum_{i=1}^{n}\|\Gamma^{obs}(t_i) - \Gamma^{obs}(t_{i-1})\|_2\right] \tag{103}$$

$$l_2 = \frac{1}{n}\sum_{i=1}^{n}\|\Gamma^{k-1}(t_i) - \Gamma^{obs}(t_i)\|_2 \tag{104}$$

where $t_0 := t_n$.

7.3 Optimal gain computation

Introducing the penalty to the objective function requires some modifications to be made to the learning algorithm. Recall that the residual error is quadratic in the gain given the remaining formlet parameters $\{x_0, y_0, \sigma, \kappa, \theta_0\}$. Therefore, the optimal gain can be found analytically. An elegant property of the energy penalty term is that it is also quadratic in the gain. The sum of quadratic functions is a quadratic function, and therefore, the regularized objective function is still quadratic in the gain given the remaining parameters. However, the expression for the optimal gain needs to be updated to reflect the new term. This can be easily done by differentiating with respect to α and setting to zero:

$$0 = \frac{\partial}{\partial\alpha}E(\Gamma^{obs}, f(\Gamma^{k-1}; \mathbf{p})), \quad \text{where } \mathbf{p} = (x_0, y_0, \sigma, \alpha, \kappa, \theta_0) \tag{105}$$

$$= \frac{\partial}{\partial\alpha}\left[F(\Gamma^{obs}, f(\Gamma^{k-1}; \mathbf{p})) + \lambda G(\mathbf{p})\right] \tag{106}$$

$$= \frac{\partial}{\partial\alpha}\left[L\|f(\Gamma^{k-1}; \mathbf{p}) - \Gamma^{obs}\|_2^2 + \lambda C\alpha^2\sigma^2 I_0(2\kappa)e^{-2\kappa}\right] \tag{107}$$

Let $g(\mathbf{x}-\mathbf{x_0}) := \dfrac{\mathbf{x}-\mathbf{x_0}}{r}A(\theta)R(r)$ (Equation 35). The residual term can then be expressed as:

$$\|\Gamma^{obs} - f(\Gamma^{k-1};\mathbf{p})\|_2^2 = \|\Gamma^{obs} - \Gamma^{k-1} - \alpha g(\Gamma^{k-1}-\mathbf{x_0})\|_2^2 \tag{108}$$

$$= \|\Gamma^{res} - \alpha g(\Gamma^{k-1}-\mathbf{x_0})\|_2^2, \quad \text{where } \Gamma^{res} := \Gamma^{obs} - \Gamma^{k-1} \tag{109}$$

$$= \|\Gamma^{res}\|_2^2 + \alpha^2\|g\|_2^2 - 2\alpha\langle\Gamma^{res}, g\rangle \tag{110}$$

where $g := g(\mathbf{x}-\mathbf{x_0})$. Therefore,

$$0 = \frac{\partial}{\partial\alpha}\left[L\|\Gamma^{res}\|_2^2 + L\alpha^2\|g\|_2^2 - 2L\alpha\langle\Gamma^{res}, g\rangle + \lambda C\alpha^2\sigma^2 I_0(2\kappa)e^{-2\kappa}\right] \tag{111}$$

$$= 2\alpha L\|g\|_2^2 - 2L\langle\Gamma^{res}, g\rangle + 2\lambda C\alpha\sigma^2 I_0(2\kappa)e^{-2\kappa} \tag{112}$$

$$= \alpha\left[\|g\|_2^2 + \frac{\lambda}{L}C\sigma^2 I_0(2\kappa)e^{-2\kappa}\right] - \langle\Gamma^{res}, g\rangle \tag{113}$$

Thus, the optimal gain can be expressed as:

$$\alpha^* = \frac{\langle\Gamma^{res}, g\rangle}{\|g\|_2^2 + \frac{\lambda}{L}C\sigma^2 I_0(2\kappa)e^{-2\kappa}} \tag{114}$$

Note that for $\lambda = 0$, the optimal gain reverts to the expression for the unregularized model (Equation 27):

$$\alpha^* = \frac{\langle\Gamma^{res}, g\rangle}{\|g\|_2^2} \tag{115}$$

As λ increases, the denominator increases so the optimal gain α^* decreases. Assuming all other variables are fixed, α^* is lower for larger scales σ and lower concentrations κ.

7.4 Selecting the mixing parameter λ

Next, I address the question of how to select the regularization parameter λ. Cross-validation is used, with a criterion that is a compromise between the convergence rate and the formlet energy. In particular, I find the largest value of λ that satisfies an upper bound on the convergence rate:

$$\lambda^{opt} = \max\{\lambda : \xi_\lambda^k \le \xi_{\lambda_0}^{k-1}\} \tag{116}$$

where ξ_λ^k is the Hausdorff error at iteration k with regularization parameter λ, and $\xi_{\lambda_0}^{k-1}$ is the value at iteration $k-1$ with no regularization ($\lambda = 0$). In other words, the strongest penalty is selected whose error

is no more than one iteration behind the unregularized model.

I tested 32 values of λ ranging over the interval [0.0002, 0.006] on a subset of of 128 shapes randomly selected from the dataset. The interval was chosen with a coarse-to-fine approach (e.g. I first tried a few wider intervals but with a coarser grid). The optimal value found was $\lambda^{opt} = 0.003942$.

7.5 Evaluation

The regularized model is evaluated on the shape reconstruction task. Figure 14 shows the pursuit of a rhino shape for the unregularized and regularized models. Figure 13 shows the composition of the formlet sequence applied to the grid. It appears that the regularized formlets are much more localized. They do not affect points that are distant from the shape, and each formlet affects a small subset of the points on the shapes. There is a clear association between each formlet and a part of the shape. At the same time, reconstruction is not substantially slowed.

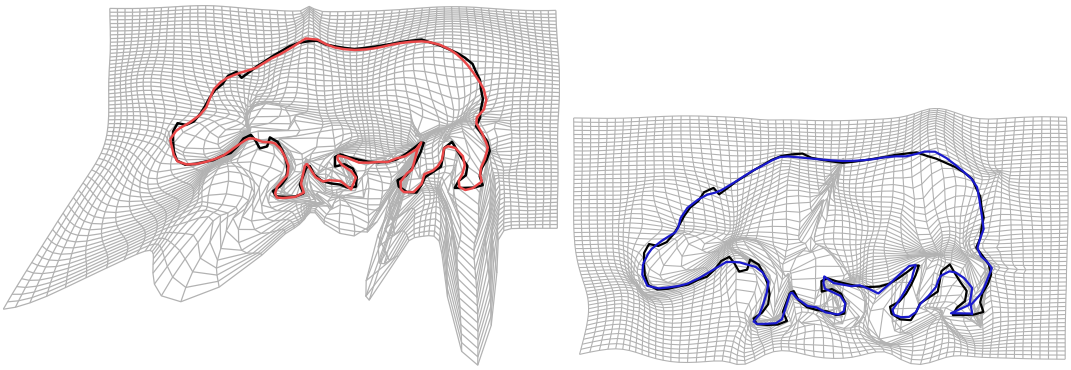


Figure 13: Formlet composition $f_{32} \circ \ldots \circ f_2 \circ f_1$ applied to the ellipse initialization and the grid. The unregularized and regularized models are shown on the left and right respectively.

To quantify these observations and see if they generalize, the convergence rates are measured across the entire dataset (Figure 15). As expected, regularizing decreases convergence accuracy, but only by a small amount. To see the effect of regularization on the formlets selected, one can consider the energy of the formlets at each iteration, that is, the quantity $\int_{\mathbb{R}^2} \|f(\mathbf{x}) - \mathbf{x}\|_2^2 d\mathbf{x}$ (Figure 16). As expected, regularizing yields a substantial decrease in the formlet energy. Regularization also affects the distributions of the scale and gain parameter (Figure 17).

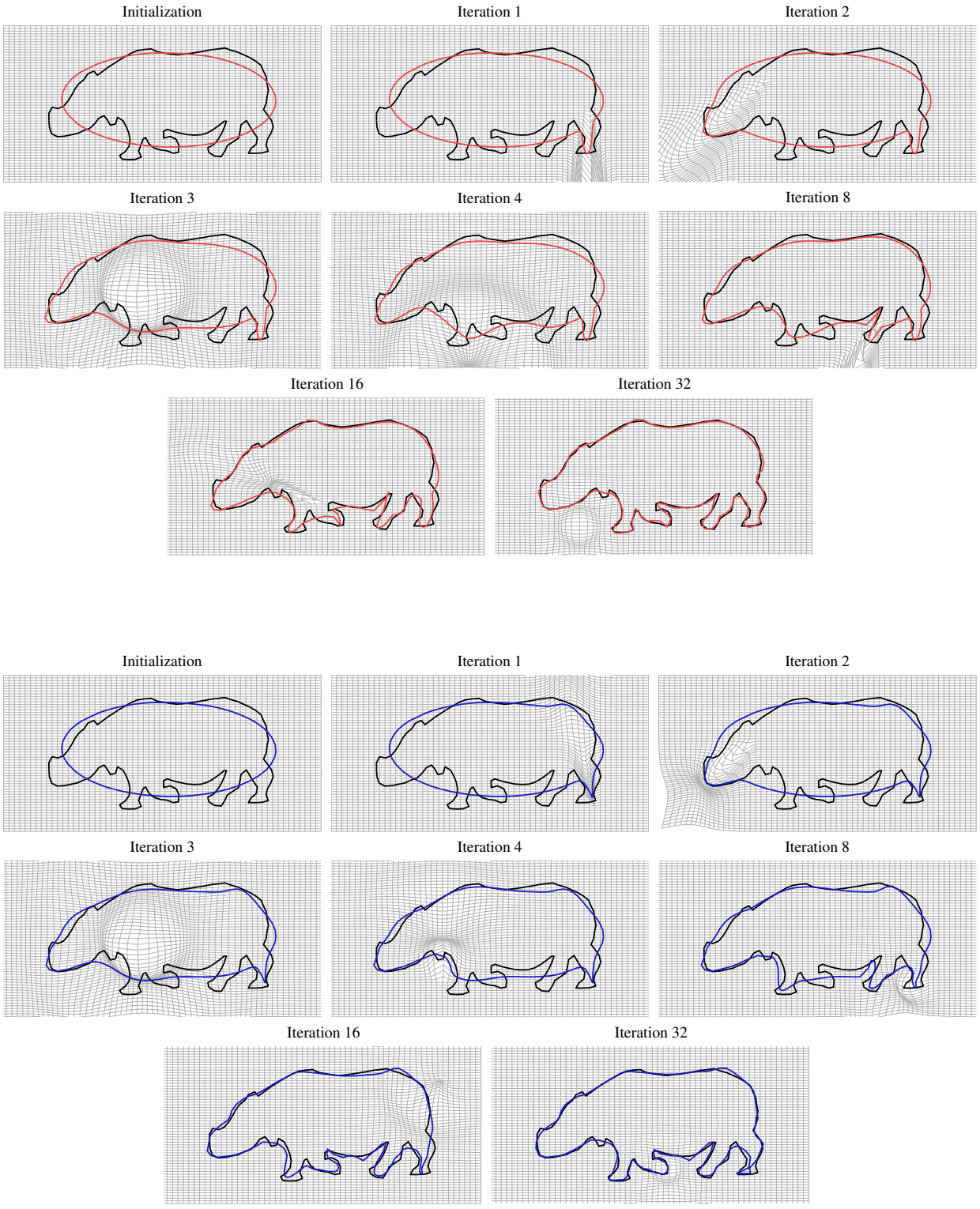


Figure 14: Pursuit of rhinoceros shape with unregularized (top) and regularized (bottom) formlet systems. Iterations 0,1,2,3,4,16,32 are shown.

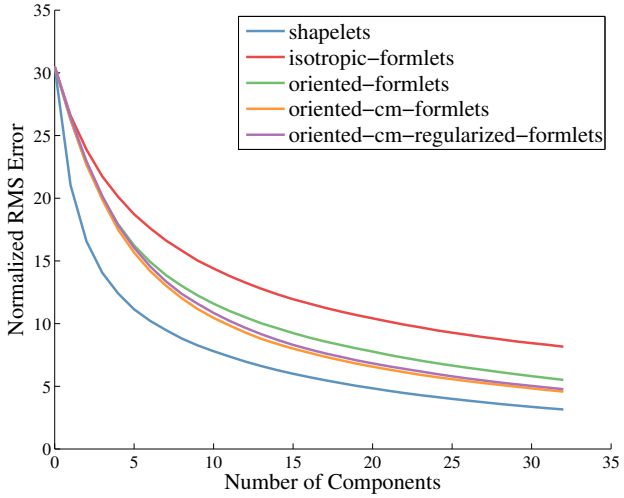


Figure 15: Reconstruction error for regularized and unregularized models.

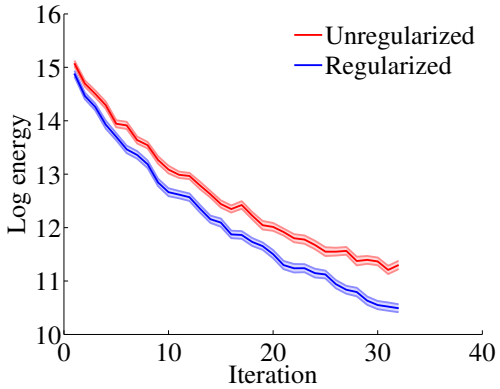


Figure 16: Log energy for regularized and unregularized models. Error bars indicate standard error of the mean.

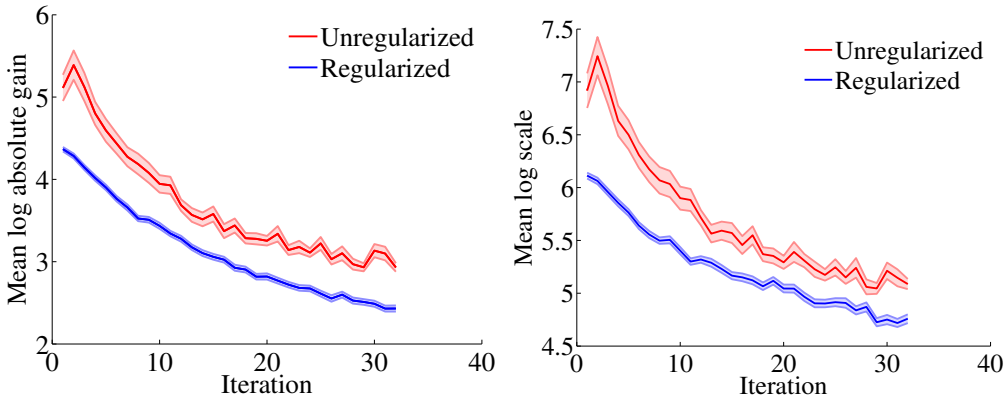


Figure 17: Distributions of gain and scale parameters for regularized and unregularized models. Error bars indicate standard error of the mean.

8 The contour completion task

The extensions of the formlet system presented thus far lead to faster shape reconstruction and improved identifiability. Next, I examine how these innovations impact the model's ability to complete fragmented shapes, an important component of perceptual organization in cluttered scenes. To simulate the completion problem, a continuous segment of each animal contour (10%, 30%, or 50%) has been selected at random for occlusion. The formlet representation is then inferred using only the remaining data (50-90%). Finally, I evaluate the error on the occluded portion using the $\mathcal{L}^2$ Hausdorff error (Equation 4.2).

8.1 Local methods for contour completion

The formlet model and its generalizations will be evaluated against shapelets. These generative models will also be compared against local methods that are not part of a complete global model of shape. Two such methods are linear interpolation (that is, simply joining the two endpoints of the visible contour with a line segment), and Elastica, which minimizes a weighted sum of squared curvature and arc length [25], subject to the constraint that the tangent orientations at the endpoints are matched to the visible curve. Bruckstein et al. proposed an algorithm for solving the following discrete version of this constrained minimization problem [7]. The energy functional is written as:

$$\int (\kappa^2 + \lambda) ds \approx \sum_{i=1}^{N} \left[\left(\frac{\theta_i}{l} \right)^2 + \lambda \right] \tag{117}$$

where κ is the curvature, θ_i is the turning angle at the i^{th} point on the curve, l is the distance between consecutive points on the curve, and λ is a free parameter. Figure 18 illustrates the discrete approximation. Following Bruckstein's notation, the curve is specified by a set of points $\{\mathbf{P_i}\}_{i=0}^{N}$. Letting Ψ_i be the angle of $\overline{P_i P_{i+1}}$ with respect to the x-axis, the turning angle at P_i is defined as $\theta_i = \Psi_i - \Psi_{i-1}$. Note that Ψ_0 and Ψ_n are fixed. Using Lagrange multipliers, the constrained optimization problem is framed as a system of nonlinear equations, solved using gradient descent [7]. The parameter λ controls the relative penalties on

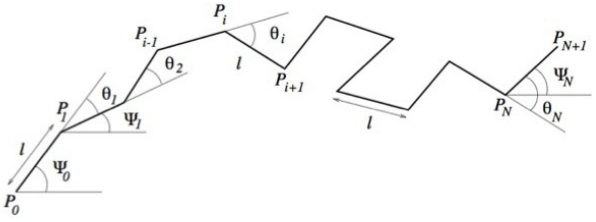


Figure 18: Elastica completion.

arc length and curvature. It has been selected by minimizing Hausdorff error using cross-validation. It was

found that $\lambda = 2.5714$ gives the lowest error on a training set of 50 animal shapes separate from the 252 shape dataset used throughout this work.

8.2 Results

Results on the completion task are shown in Figure 19. Several observations can be made:

1. All formlet systems perform better than shapelets on the completion task.
2. Of all generative models, isotropic formlets seem to be best at the completion.
3. Linear interpolation outperforms all generative models at 10% occlusion, but its accuracy deteriorates as occlusion increases. At 30% occlusion it is on par with formlets, and at 50% it is much worse.
4. Elastica provides the most accurate completions. This is likely due in part to the fact that it is the only method that was trained on a dataset of natural shapes, and is the only method that explicitly constraints the completion to be smooth. Note that for larger (50%) occlusions, isotropic formlets are becoming competitive with elastica.

I expect that the performance of the formlet system could be further improved by learning a probabilistic models over formlet parameters.

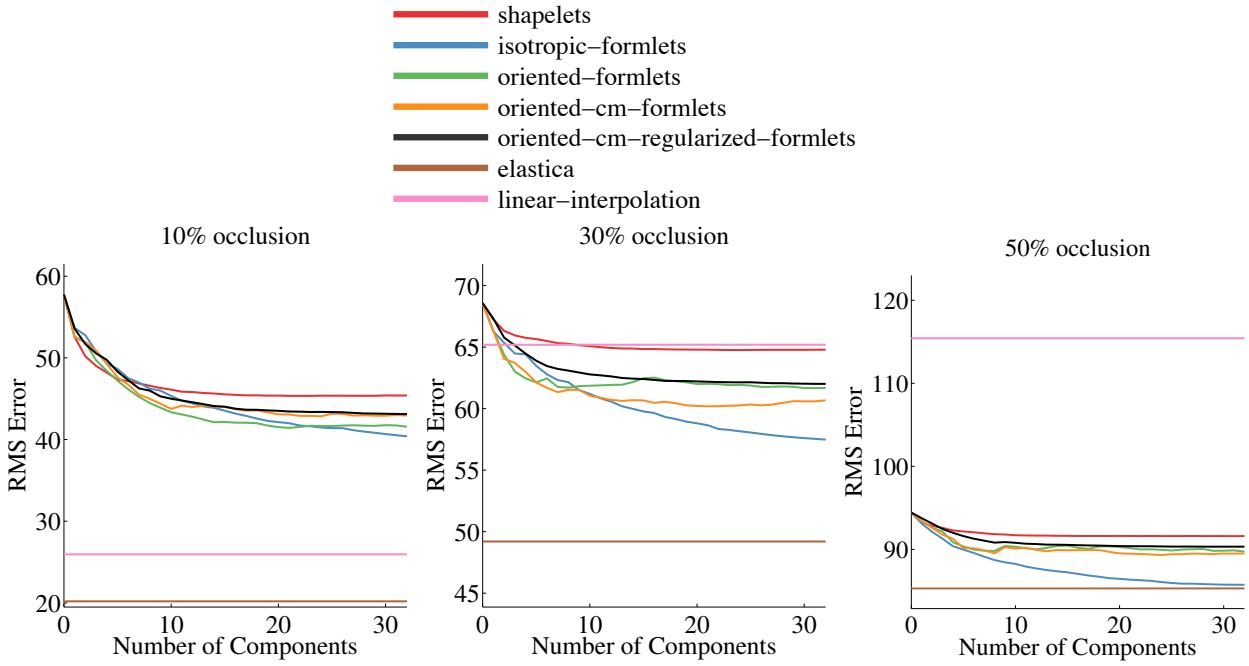


Figure 19: Quantitative comparison of contour completion methods.

9 Grouping formlets into parts

9.1 Motivation

This chapter addresses the problem of structuring the formlet representation by clustering the model elements into parts. The current sequential structure assumed by the model implies that every formlet depends on its predecessor in the sequence, which does not reflect the hierarchical part structure present in animal shapes. The goal is to relax the ordering of formlets to a partial ordering.

The strengths of a structured part-based representation are several. First, it should increase the robustness of the model to occlusion and noise. If a part of the shape is missing or corrupted, only the part of the model encoding that part will be affected. Second, many objects seem to have a natural part-based structure, and so this representation would be in good correspondence with the natural structure of the object. Third, it is hoped that such a representation will be more convenient for articulated objects, where articulations may occur between rigid parts. Finally, such a representation leads to a natural sparse graphical model that will facilitate probabilistic modelling.

A probabilistic model defines a joint probability distribution over a set of random variables, in our case, the sequence of formlets $\{f_1, f_2, \ldots f_{32}\}$ encoding a given shape. If every formlet to depends on every other, there will be too many parameters in the model. Therefore, for learning to be tractable, some independence relationships between formlets must be identified using a graphical model. Building a probabilistic graphical model is outside the scope of this thesis, so I only consider the problem of learning a structure which I conjecture may be an appropriate graphical model.

9.2 Candidate graphical models

There are several graph structures to consider, each making different independence assumptions. These are shown in Figure 20 and enumerated below.

1. *Markov chain* - assume a formlet depends only on its predecessor in the sequence, that is:

$$f_i \perp f_1, f_2 \ldots f_{i-2} \quad | f_{i-1} \quad \forall i = 1 \ldots K \tag{118}$$

   This condition describes a first order Markov chain. In a k^{th} order Markov chain, the dependence is on the last k predecessors.

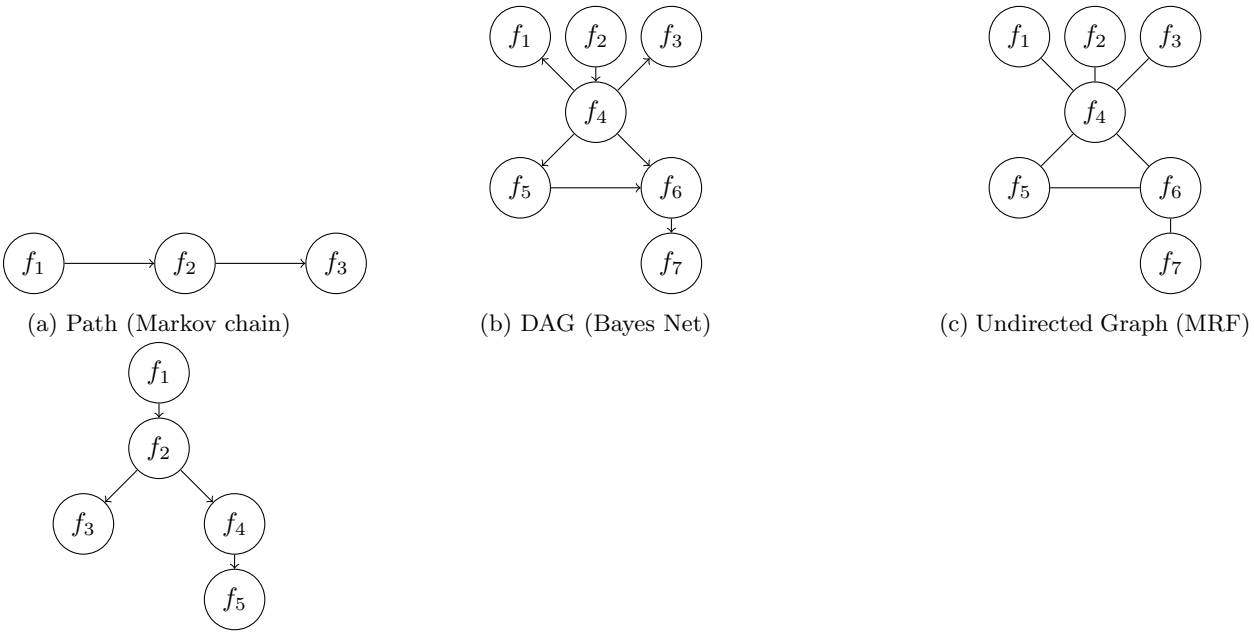
2. *Bayes Net* - the underlying graph is a directed acyclic graph (DAG), in which a formlet only depends

on its parent(s):

$$f_i \perp f_j \quad |\mathbf{Pa}(f_i) \quad \forall i, j = 1 \ldots K, \quad j \notin \mathbf{Pa}(f_i) \tag{119}$$

where $\mathbf{Pa}(f_i)$ denotes the set of parents of f_i.

3. *Markov Random Field (MRF)* - independence assumptions are captured by any undirected graph (cycles are allowed).



(a) Path (Markov chain) (b) DAG (Bayes Net) (c) Undirected Graph (MRF)

(d) Rooted Tree

Figure 20: Candidate graphical models.

The Markov Chain model is not appropriate, since it does not allow multiple sub–parts of a common parent (e.g. the fingers of a hand). The undirected nature of the MRF model is arguably not suited for the structure of the formlet relationships, as it fails to capture any temporal ordering in the growth process or hierarchy in the part structure.

This leaves the Bayes Net, which encompasses a fairly broad class of models. However, out of the space of all directed acyclic graphs, only rooted trees will be considered. A rooted tree is any acyclic graph where one node is designated as the root, and all edges have an orientation towards/away from this root. Rooted trees have the property that every node (except the root) has a unique parent, defined as its neighbour on the path to the root. This simplification is justified by noting the hierarchical nature of animal shapes – a hand is a subpart of an arm, digits are subparts of a hand, and so on. This is consistent with the view of a given shape as a growth process in which every part has grown from exactly one direct ancestor.

Next I address the problem of computing the tree structure given a formlet sequence. This can be treated as a hierarchical clustering problem in which the association between pairs of formlets is measured and the parent of formlet f_i is defined as the formlet $f_j, \quad j < i$, maximizing association. The problem is constrained by the assumption that a parent formlet must precede a child in the formlet sequence. To apply this rule, a metric for the association between formlets is needed.

9.3 Metric for Formlet Association

The naive approach would be to work directly with the deformation parameters, but this is complicated by the different interactions between the location, scale, gain, orientation and concentration. For example, two formlets that are close in space may encode very different parts of the shape depending on their orientation and concentration. The real quantity of interest is the effect of a formlet on the model curve. If two formlets affect different parts of the shape, they should be less likely to be neighbours on the graph. The effect of a formlet can be measured by the magnitude of the displacement of points along the curve. Let the action of a formlet f be defined as the vector $\mathbf{a} = (a_1, a_2, \ldots, a_n)^T$, where $a_i = \|\Gamma^k(t_i) - \Gamma^{k-1}(t_i)\|_2$. Figures 22 and 25 show the formlet actions for two animal shapes.

Association (normalized correlation) between formlets f_i and f_j can be measured by the angle between their action vectors:

$$C(f_i, f_j) = \frac{\mathbf{a_i}^T \mathbf{a_j}}{\|\mathbf{a_i}\|_2 \|\mathbf{a_j}\|_2} \tag{120}$$

Note that since the formlet actions are nonnegative, it follows that $0 \leq C(f_i, f_j) \leq 1 \quad \forall i, j$. The association is zero if the two action vectors do not overlap, and one if they are collinear.

The normalized correlation measures the relative orientation of the two action vectors, ignoring their magnitudes. This is desirable, since due to the coarse-to-fine nature of the pursuit, there is much variation in the scale and gain. Without normalizing, large-scale formlets that appear early on in the pursuit would dominate all the other formlets. The correlation structure can visualized as a symmetric $K \times K$ matrix, shown in Figure 21 for an example shape. This matrix is fairly sparse - for any given formlet, the correlation exceeds 0.5 for only a few other formlets.

9.4 Re-parameterization

One complication is that the model curve is re-parameterized at every iteration of the pursuit. While a one-to-one correspondence between points on the model curve is maintained, it needs to be updated at each iteration since points are added and removed. The addition and removal of points can be tracked

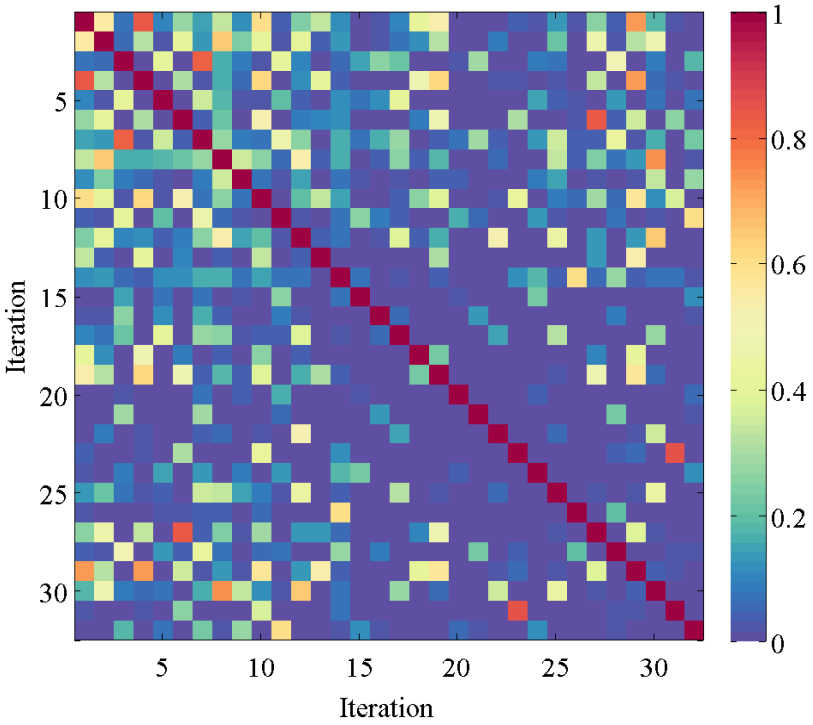


Figure 21: Correlation matrix for the rhino shape.

with a $n \times n$ matrix M, where $n = 128$ is the number of points defining each curve. Consider an example with the following associations between the points on the model curve before and after re-parameterization: $\{a_1 \to b_1, a_2 \to b_2, a_2 \to b_3, a_3 \to b_4, a_4 \to b_4\}$ where the a_i are points on the old curve and b_i are points on the new curve. The matrix M storing these associations is:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

Each column of this matrix encodes the coordinates of a new point in terms of the old points. When comparing the action of two formlets, say $\mathbf{a_i}$ and $\mathbf{a_j}$, the action vectors should be the same coordinate system. Assuming without loss of generality that $i < j$, $\mathbf{a_j}$ is transformed by premultiplying by the composition of the mapping matrices for each iteration:

$$\mathbf{a_i} = (M_i M_{i+1} \dots M_j)\mathbf{a_j} \tag{121}$$

The *total action* of a formlet with action vector $\mathbf{a} = (a_1, a_2, \ldots a_n)$ can be defined as the scalar $\sum_{i=1}^{n} a_i$. The total action of a formlet should be not change when the coordinates change. It can be easily shown that as long as the columns of M sum to 1, the total action will be invariant under this linear transformation:

Proposition 2. *Suppose* $\mathbf{v} = M\mathbf{u}$*, where* $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ *and* M *is a* $n \times n$ *matrix whose every column sums to 1. Then* $\sum_{i=1}^{n} u_i = \sum_{j=1}^{n} v_j$*.*

Proof.

$$\sum_i v_i = \sum_i \sum_j M_{ij} u_j \tag{122}$$

$$= \sum_j u_j \sum_i M_{ij}, \quad \text{where } \sum_i M_{ij} = 1 \quad \forall j = 1 \ldots n \tag{123}$$

$$= \sum_j u_j \tag{124}$$

□

9.5 Algorithm

Equipped with a measure of similarity between two formlets, I now turn to the problem of learning the tree structure for a given shape. The problem is essentially to identify the root(s) and the parent node of each of the remaining formlets.

The order that formlets appear in the sequence can be used to constrain the search space for parents. In particular, it can be assumed that the parent of a formlet must appear earlier in the sequence. This is consistent with the view of the deformation sequence as a growth process in which parts grow out of their ancestors.

To find the tree, the formlet sequence is traversed in order, defining the parent of each formlet to be the closest formlet already in the tree (Algorithm 2). The root is simply the first formlet in the sequence.

Algorithm 2: Hierarchical clustering of formlet representation

Initialization: f_1 is the root

for $k = 2, \ldots, K$ **do**

Find parent: compute closest formlet in tree:

$$\mathbf{Pa}(f_k) = \underset{f_i \in \{f_1, \ldots f_{k-1}\}}{\operatorname{argmax}} C(f_k, f_i)$$

The algorithm is conceptually simple and easy to implement. A MATLAB implementation on standard hardware takes around 60 milliseconds to run with $K = 32$ formlets.

9.6 Evaluation

Results of the proposed part structure algorithm are shown for a sample of two animal shapes. There are several ways to visualize the tree decompositions. Figures 22 shows the pursuit of the rhino shape, illustrating the formlet action at each iteration. Figure 23 shows the topology of the part tree, as well as the the association between every pair of nodes. Figures 25 and 26 show the corresponding results for the goat shape.

A good graphical model should decompose into subgraphs that encode independent parts of an animal shape (e.g. head, limbs). In the formlet model, a part corresponds to a complete subtree. For example, for the rhino shape, the subtree with nodes $\{2, 18\}$ encodes the hind leg. The other subtrees encode the head/front leg, backside, crotch, and belly/middle legs. The correspondence between subtrees and parts of the shape can be seen more clearly by applying only the formlets in a given subtree and its ancestors. Figures 24, 27 illustrate the part subtrees for the two example shapes. It appears that subtrees encode localized regions of the shapes, and that there is a rough correspondence between the subtrees and what humans may perceive as parts. However, this correspondence is far from perfect; for example, the subtree rooted at formlet 3 encodes a collection of three related parts: the leg, head, and horn of the rhino.

A complete quantitative evaluation of the clustering algorithm is beyond the scope of this thesis. To qualitatively evaluate the model, I measure the independence of two sets of formlets by their commutativity. That is, two formlets f_A and f_B are independent if the order in which they are composed doesn't affect the outcome:

$$f_A \circ f_B(\Gamma_k) = f_B \circ f_A(\Gamma_k)$$

To assess subtree independence, I compare samples generated from two sequences of formlets, one obtained by randomly permuting the original sequence, and the other by randomly permuting the formlets, while respecting the partial ordering induced by the tree. These permutations are shown in Figures 28 and 29 for the rhino shape and Figures 30 and 31 for the goat shape. If the tree accurately reflects independence properties of the formlets, the permutations that respect the partial ordering should yield higher fidelity reconstructions. Figures 28, 30 show that this is indeed the case.

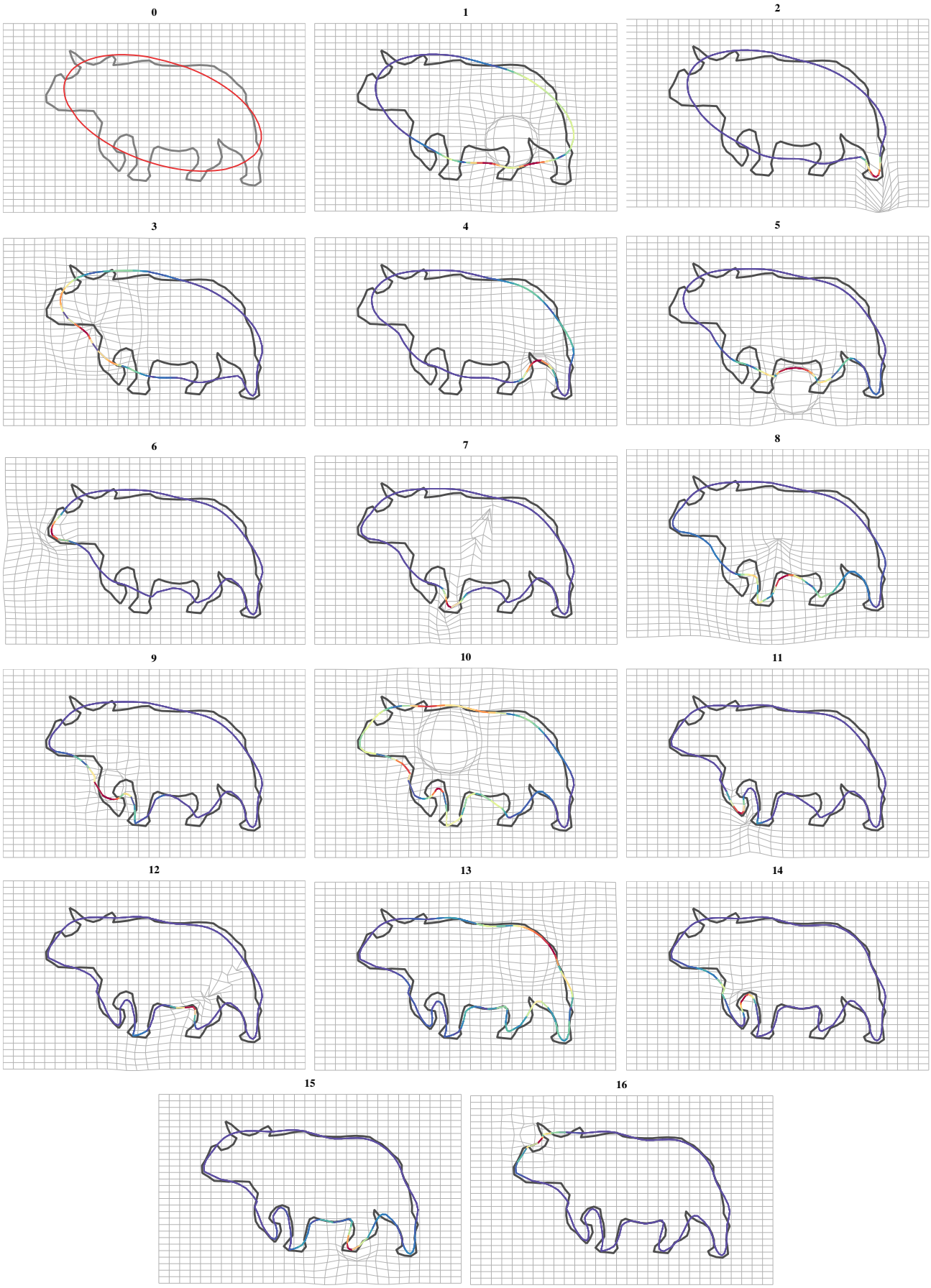


Figure 22: Pursuit of the rhino shape. The colour indicates the action of the formlet at every point on the target shape. The colour map is renormalized for every image (e.g. red indicates the maximum action at every iteration).

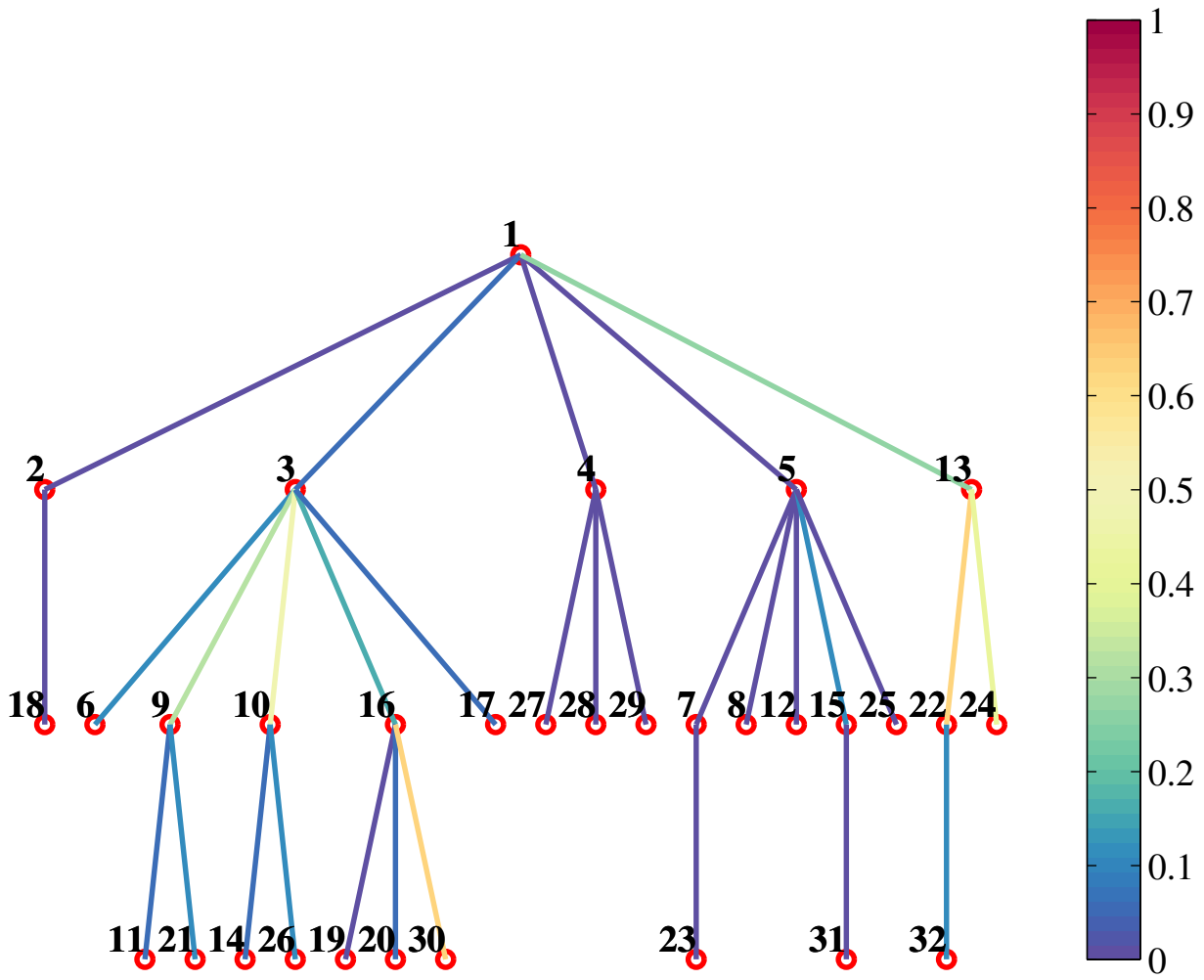


Figure 23: Part tree for the rhino shape. Edge colour represents the strength of the parent-child relationship.

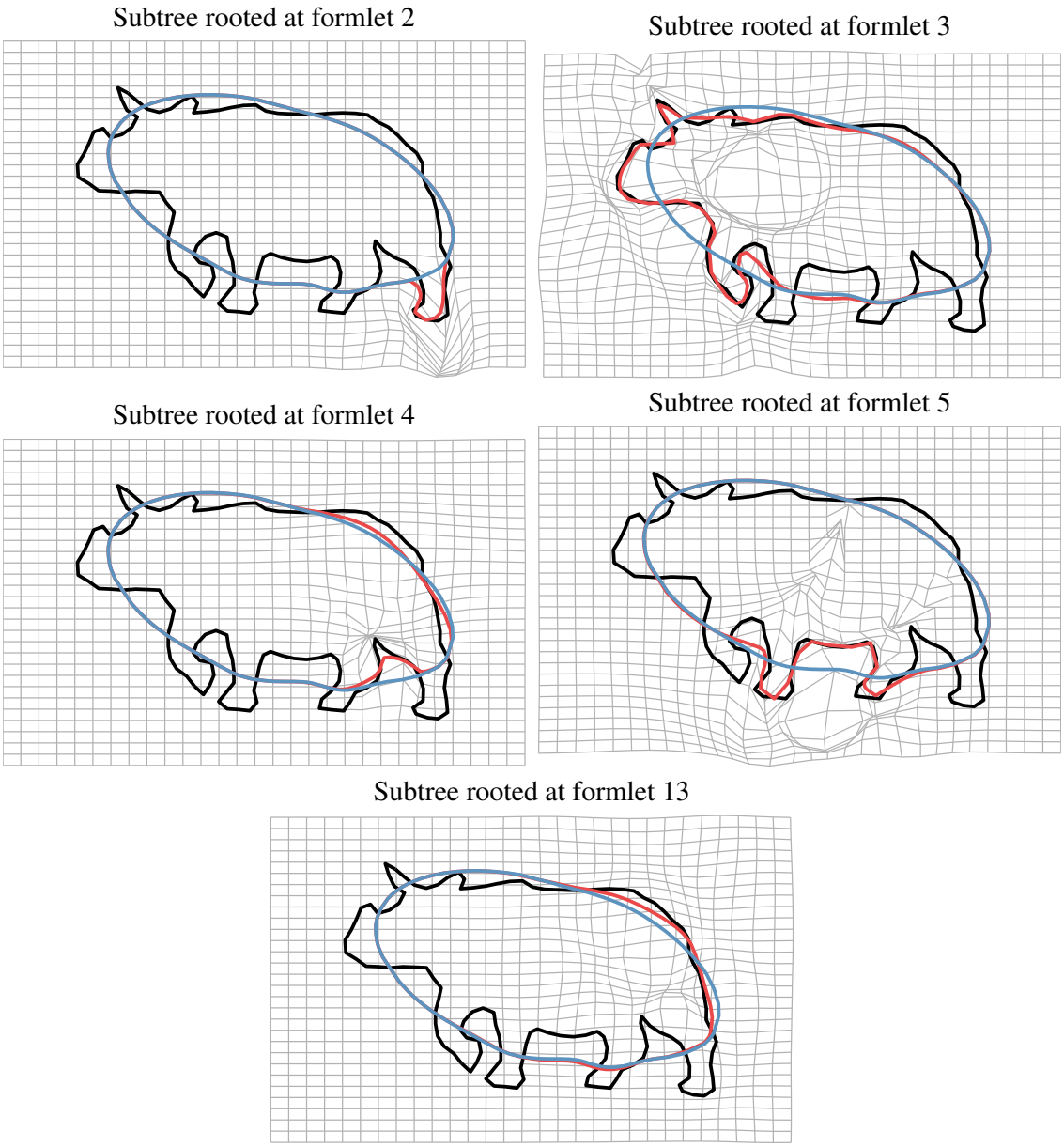


Figure 24: Automatically computed parts for the rhino shape. Each image shows the subtree root formlet (blue) and the result of applying the formlet subtree on the subtree root shape (red).

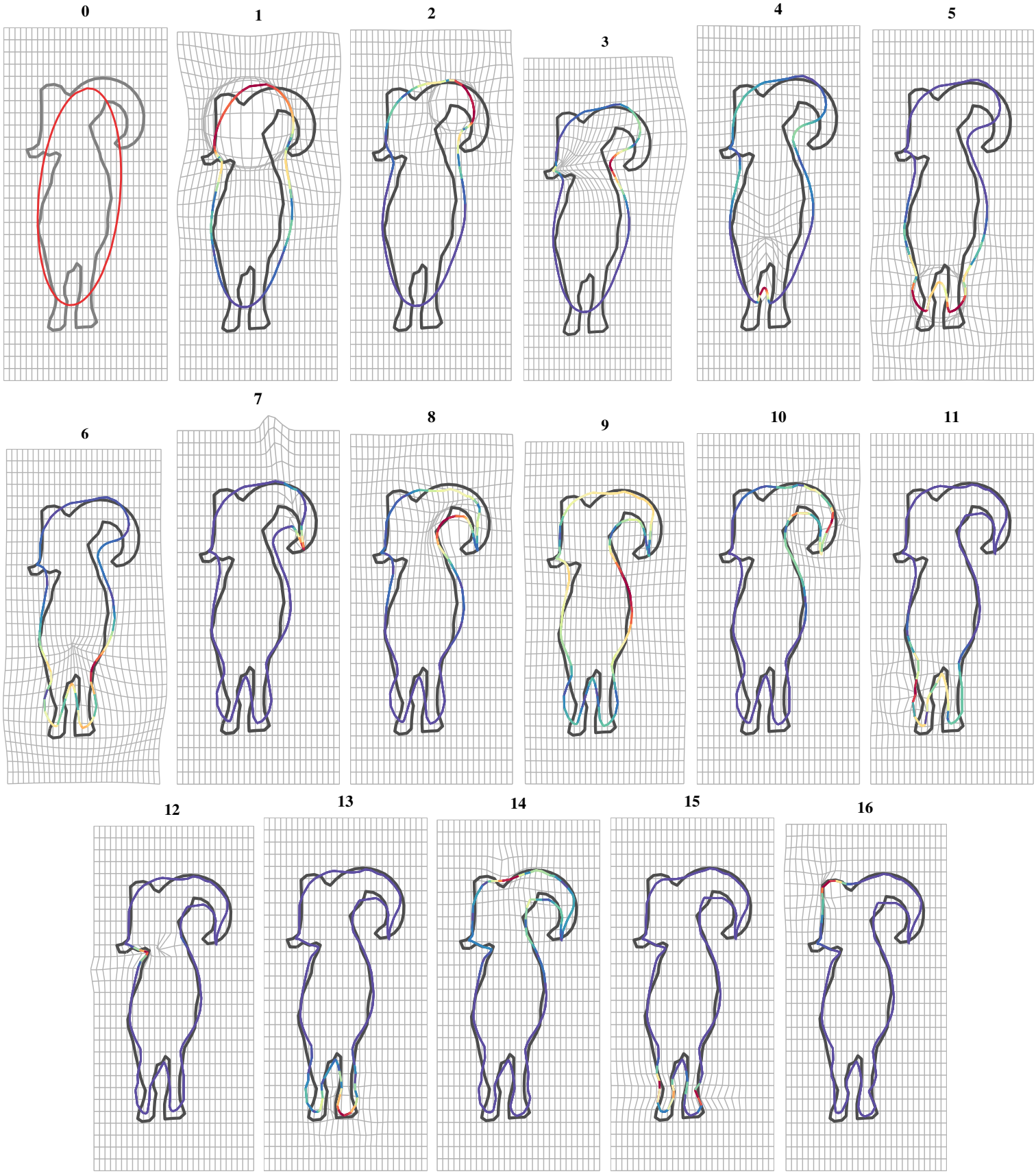


Figure 25: Pursuit of the goat shape.

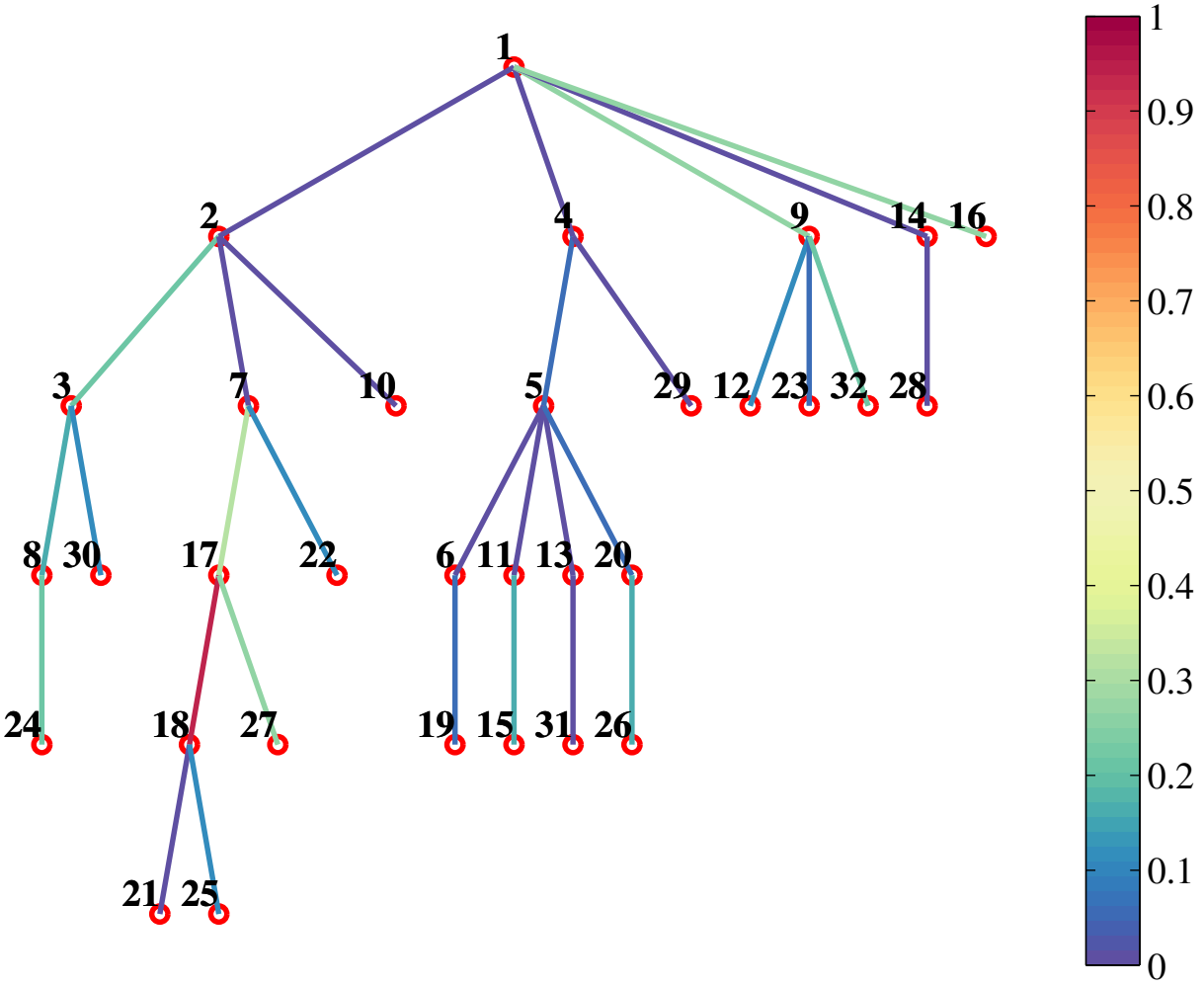
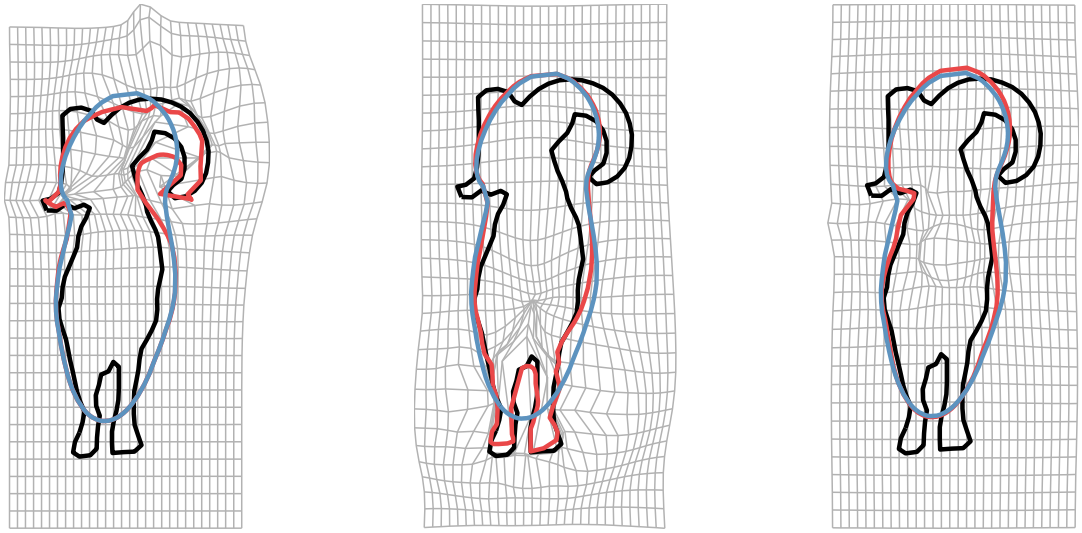


Figure 26: Part tree of the goat shape.

Subtree rooted at formlet 2 Subtree rooted at formlet 4 Subtree rooted at formlet 9



Subtree rooted at formlet 14 Subtree rooted at formlet 16

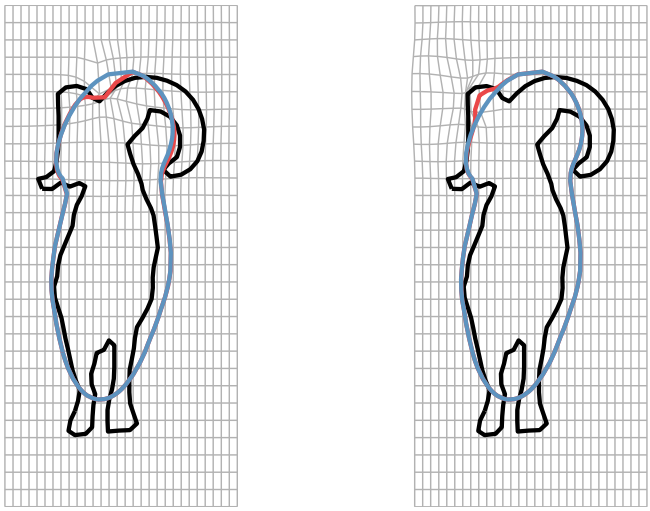


Figure 27: Parts of the goat shape.

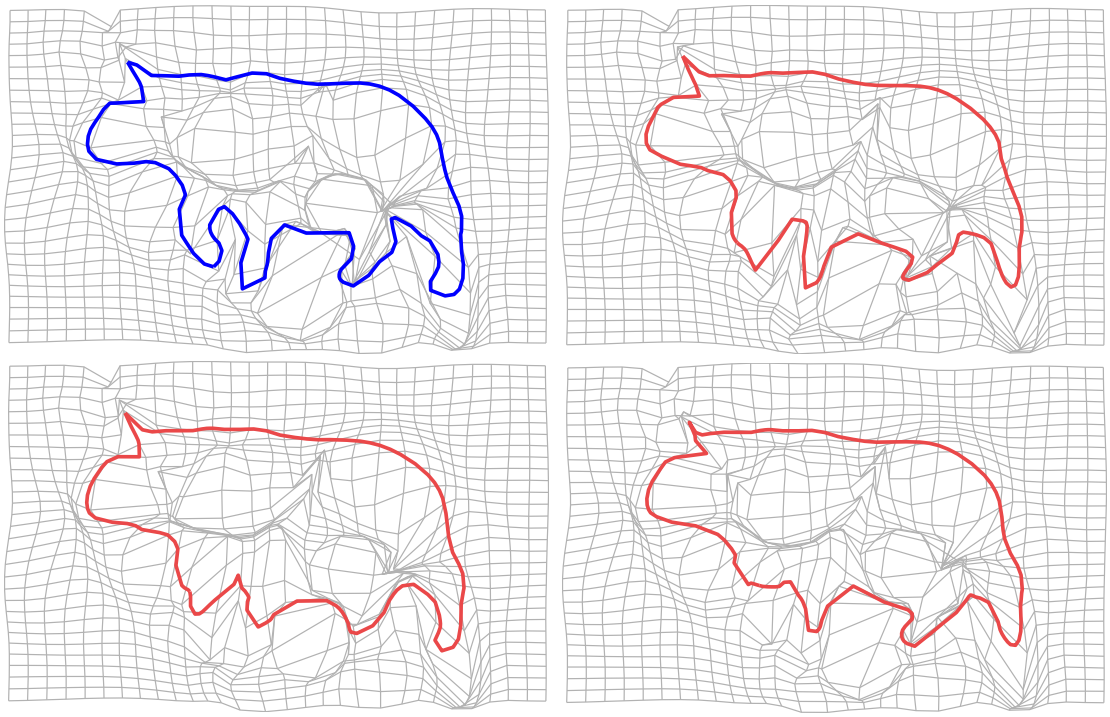


Figure 28: Samples obtained by randomly permuting the sequence of formlets $f_1 \dots f_{32}$. Composition of the original sequence is shown in blue.

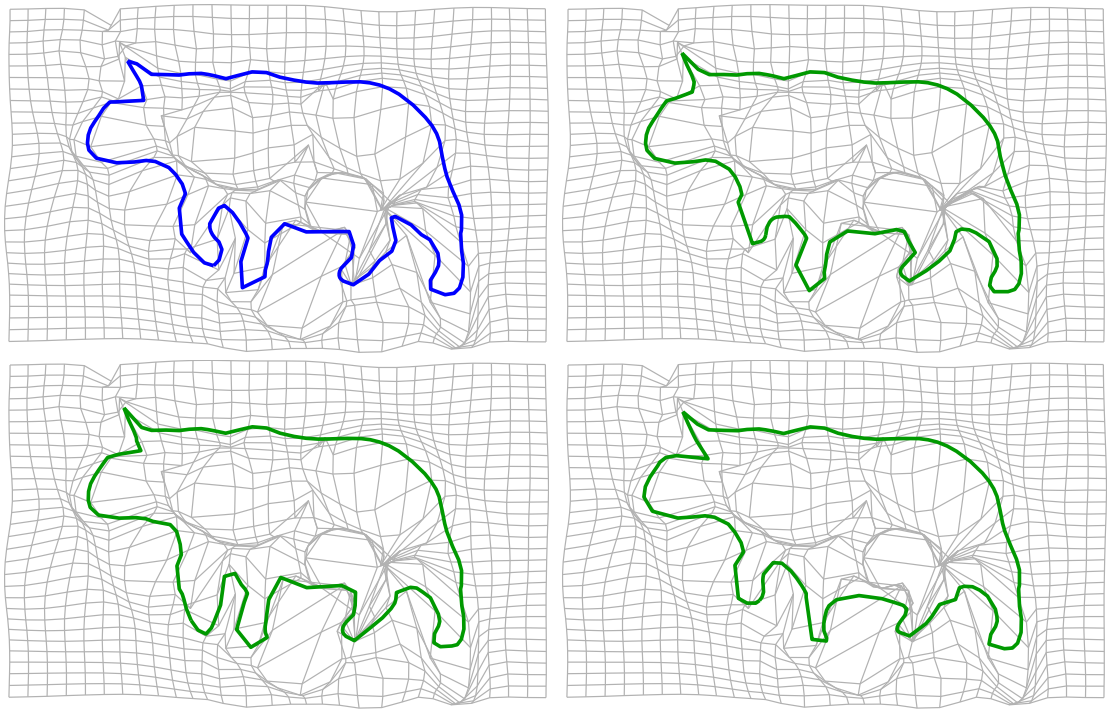


Figure 29: Samples obtained by randomly permuting formlets, while preserving the partial ordering induced by the part tree. Composition of the original sequence is shown in blue.

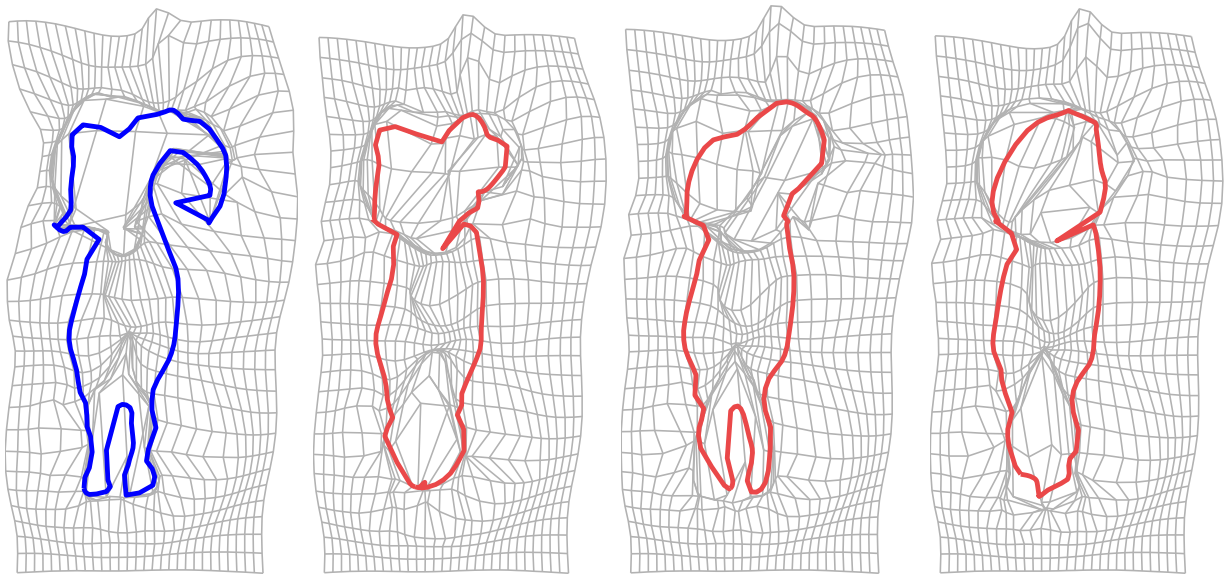


Figure 30: Samples obtained by randomly permuting the sequence of formlets $f_1 \ldots f_{32}$. Composition of the original sequence is shown in blue.

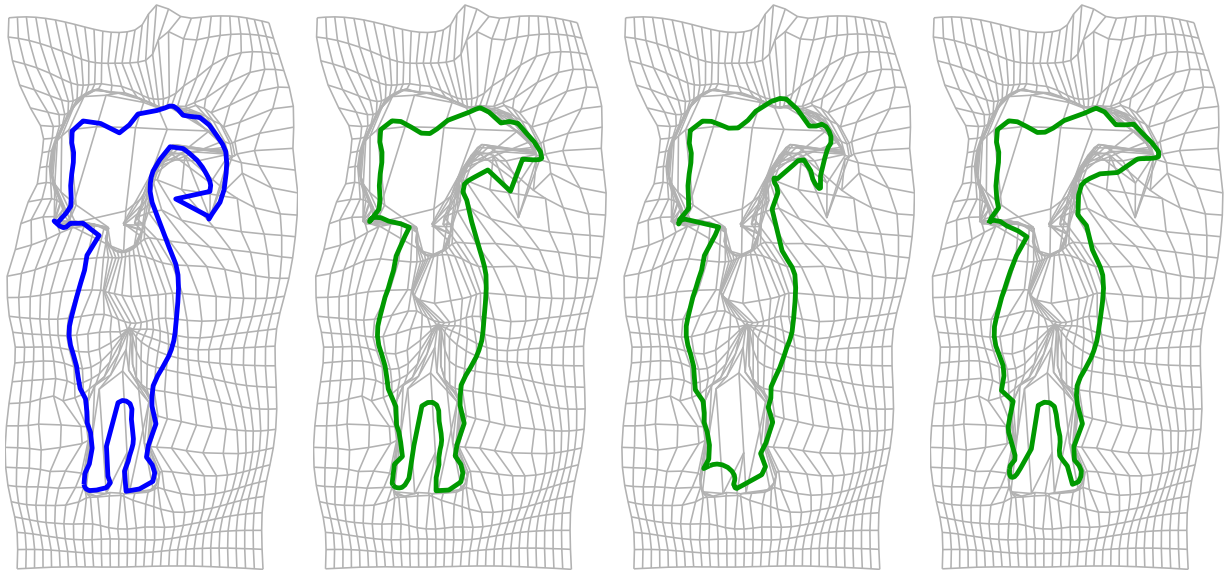


Figure 31: Samples obtained by randomly permuting formlets, while preserving the partial ordering induced by the part tree. Composition of the original sequence is shown in blue.

10 Conclusion

The formlet system is a powerful way to represent shapes and, unlike competing methods, it satisfies properties essential to a shape model. However, it suffers from several limitations that must be overcome before it can become useful in practice. This thesis presented several contributions that address these limitations.

First, I demonstrated that generalizing the formlet basis to include oriented deformations helps capture the elongated features ubiquitous in natural shapes, and speeds the reconstruction on average. I introduced a novel procedure for maintaining a natural correspondence between the target and model shapes, further improving convergence. Thirdly, I showed that regularizing in the data space can effectively dampen the scale and gain of the selected formlets, improving the identifiability of the model, without substantially impacting the convergence. Finally, I explored a graphical model that captures the hierarchical nature of animal shapes and induces a partial ordering on the representation.

A natural next step would be to build a probabilistic model of formlets. In its current state, the formlet system is not a generative model per se, but only the foundation for such a model. It remains to learn the joint probability distribution over observed shapes and formlet parameters so that the model could 'hallucinate' different shapes. This may improve completion performance.

Once the formlet model is complete, it can be applied to a variety of vision problems for which shape is an important feature. For example, it can be incorporated in a perceptual grouping framework to help guide the grouping of contour fragments in a top-down fashion.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[2] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183, 1954.

[3] C. Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006.

[4] H. Blum. Biological shape and visual science (part I). *Journal of Theoretical Biology*, 38(2):205–287, 1973.

[5] H. Blum and R.N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180, 1978.

[6] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61, 1984.

[7] A.M Bruckstein, R.J Holt, and A.N Netravali. Discrete elastica. *Applicable Analysis*, 78(3-4):453–485, 2001.

[8] G. Casella and R.L Berger. *Statistical inference*, volume 70. Duxbury Press Belmont, CA, 1990.

[9] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, et al. Active shape models – their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[10] I.L Dryden and K.V Mardia. *Statistical shape analysis*. Wiley, 1998.

[11] A. Dubinskiy and S.C Zhu. A multi-scale generative model for animate shapes and parts. *Proceedings of the 9th IEEE ICCV*, 1:249–256, October 2003.

[12] J.H Elder, T.D Oleskiw, A. Yakubovich, and G. Peyré. On growth and formlets: Sparse multi-scale coding of planar shape. *Image and Vision Computing*, 2012.

[13] J. Feldman and M. Singh. Bayesian estimation of the shape skeleton. *Proceedings of the National Academy of Sciences*, 103(47):18014–18019, 2006.

[14] N. Fisher. *Statistical analysis of circular data*. Cambridge University Press, 1995.

[15] U. Grenander, A. Srivastava, and S. Saini. Characterization of biological growth using iterated diffeomorphisms. In *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pages 1136–1139. IEEE, 2006.

[16] U. Grenander, A. Srivastava, and S. Saini. A pattern-theoretic characterization of biological growth. *Medical Imaging, IEEE Trans.*, 26(5):648–659, 2007.

[17] D.D Hoffman and W.A Richards. Parts of recognition. *Cognition*, 18(1):65–96, 1984.

[18] A.K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(3):267–278, 1996.

[19] M. Leyton. A process-grammar for shape. *Artificial Intelligence*, 34(2):213–247, 1988.

[20] M. Leyton. *Symmetry, causality, mind.* MIT Press, Cambridge, Mass., 1992.

[21] M. Maes. On a cyclic string-to-string correction problem. *Information Processing Letters*, 35(2):73–78, 1990.

[22] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[23] V. Movahedi and J.H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 49–56. IEEE, 2010.

[24] D. Mumford. Mathematical theories of shape: Do they model perception? In *Geometric Methods in Computer Vision*, volume 1570, pages 2–10. International Society for Optics and Photonics, 1991.

[25] D. Mumford. Elastica and computer vision. In *Algebraic Geometry and its Applications*, pages 491–506. Springer, New York, 1994.

[26] T.D Oleskiw, J.H Elder, and G. Peyré. On growth and formlets: Sparse multi-scale coding of planar shape. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 459–466, June 2010.

[27] N. Portman, U. Grenander, and E.R Vrscay. Direct estimation of biological growth properties from image data using the GRID model. In *Image Analysis and Recognition*, pages 832–843. Springer, 2009.

[28] S.J.D. Prince. *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.

[29] E. Sharon and D. Mumford. 2D shape analysis using conformal mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 350–357. IEEE, 2004.

[30] L.H Staib and J.S Duncan. Parametrically deformable contour models. In *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on*, pages 98–103. IEEE, 1989.

[31] D.W Thompson. *On growth and form*. Cambridge Univ. Press, 1942.

[32] J. Wagemans, J.H Elder, M. Kubovy, S.E Palmer, M.A Peterson, M. Singh, and R. von der Heydt. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological bulletin*, 138(6):1172, 2012.

[33] Q. Xie and A. Srivastava. An improved estimator of GRID model for representing large diffeomorphic deformations. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy-Geometrical and Statistical Methods for Modelling Biological Shape Variability*, pages 25–37, 2011.

[34] A. Yakubovich and J.H Elder. Building better formlet codes for planar shape. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 84–91. IEEE, 2014.

[35] S.C Zhu. Embedding Gestalt laws in Markov random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(11):1170–1187, 1999.